Non-Malleability from Malleability: Simulation-Sound Quasi-Adaptive NIZK Proofs and CCA2-Secure Encryption from Homomorphic Signatures

Benoît Libert, Thomas Peters, Marc Joye, Moti Yung

Non-Malleability from Malleability: Simulation-Sound Quasi-Adaptive NIZK Proofs and CCA2-Secure Encryption from Homomorphic Signatures

Benoît Libert[1] ⋆, Thomas Peters[2] ⋆⋆, Marc Joye[1], and Moti Yung[3]

[1] Ecole Normale Supérieure de Lyon, Laboratoire d'Informatique du Parallélisme (France)
[2] Technicolor (France)
[3] Université catholique de Louvain, Crypto Group (Belgium)
[4] Google Inc. and Columbia University (USA)

Abstract. Verifiability is central to building protocols and systems with integrity. Initially, efficient methods employed the Fiat-Shamir heuristics. Since 2008, the Groth-Sahai techniques have been the most efficient in constructing non-interactive witness indistinguishable and zero-knowledge proofs for algebraic relations in the standard model. For the important task of proving membership in linear subspaces, Jutla and Roy (Asiacrypt 2013) gave significantly more efficient proofs in the quasi-adaptive setting (QA-NIZK). For membership of the row space of a $t \times n$ matrix, their QA-NIZK proofs save $\Omega(t)$ group elements compared to Groth-Sahai. Here, we give QA-NIZK proofs made of a *constant* number group elements – regardless of the number of equations or the number of variables – and additionally prove them *unbounded* simulation-sound. Unlike previous unbounded simulation-sound Groth-Sahai-based proofs, our construction does not involve quadratic pairing product equations and does not rely on a chosen-ciphertext-secure encryption scheme. Instead, we build on structure-preserving signatures with homomorphic properties. We apply our methods to design new and improved CCA2-secure encryption schemes. In particular, we build the first efficient threshold CCA-secure keyed-homomorphic encryption scheme (*i.e.*, where homomorphic operations can only be carried out using a dedicated evaluation key) with publicly verifiable ciphertexts.

Keywords. NIZK proofs, simulation-soundness, chosen-ciphertext security, homomorphic cryptography.

1 Introduction

Non-interactive zero-knowledge proofs [8] play a fundamental role in the design of numerous cryptographic protocols. Unfortunately, until breakthrough results in the last decade [31–33], it was not known how to construct them efficiently without appealing to the random oracle methodology [7]. Groth and Sahai [33] described very efficient non-interactive witness indistinguishable (NIWI) and zero-knowledge (NIZK) proof systems for algebraic relations in groups equipped with a bilinear map. For these specific languages, the methodology of [33] does not require any proof of circuit satisfiability but rather leverages the properties of homomorphic commitments in bilinear groups. As a result, the length of each proof only depends on the number of equations and the number of variables.

While dramatically more efficient than general NIZK proofs, the GS techniques remain significantly more expensive than non-interactive proofs obtained from the Fiat-Shamir heuristic [26] in the random oracle model [7]: for example, proving that t variables satisfy a system of n linear equations demands $\Theta(t+n)$ group elements where Σ-protocols allow for $\Theta(t)$-size proofs. In addition, GS proofs are known to be malleable which, although useful in certain applications [5, 18], is undesirable when NIZK proofs serve as building blocks for non-malleable protocols. To construct chosen-ciphertext-secure encryption schemes [50], for example, the Naor-Yung/Sahai [46, 51] paradigm requires NIZK proofs satisfying a form of non-malleability called *simulation-soundness* [51]: informally, this property captures the inability of the adversary to prove false statements by itself, even after having observed simulated proofs for possibly false statements of its choice.

Groth-Sahai proofs can be made simulation-sound using constructions suggested in [32, 15, 34].

⋆ This work was done while this author was at Technicolor (France).
⋆⋆ This author was supported by the CAMUS Walloon Region Project.

However, even when starting from a linear equation, these techniques involve proofs for quadratic equations, which results in longer proofs. One-time simulation-soundness (*i.e.*, where the adversary only sees one simulated proof) is more economical to achieve as shown in [39, 42]. Jutla and Roy suggested a more efficient way to achieve a form of one-time simulation-soundness [37].

Quasi-Adaptive NIZK Proofs. For languages consisting of linear subspaces of a vector space, Jutla and Roy [38] recently showed how to significantly improve upon the efficiency of the GS paradigm in the *quasi-adaptive* setting. In quasi-adaptive NIZK proofs (QA-NIZK) for a class of languages $\{\mathcal{L}_\rho\}$ parametrized by ρ, the common reference string (CRS) is allowed to depend on the particular language $\mathcal{L}_\rho$ of which membership must be proved. At the same time, a single simulator should be effective for the whole class of languages $\{\mathcal{L}_\rho\}$. As pointed out in [38], QA-NIZK proofs are sufficient for many applications of Groth-Sahai proofs. In this setting, Jutla and Roy [38] gave very efficient QA-NIZK proofs of membership in linear subspaces. If $\mathbf{A} \in \mathbb{Z}_p^{t\times n}$ is a matrix or rank $t < n$, in order to prove membership of the language $\mathcal{L} = \{\boldsymbol{v} \in \mathbb{G}^n \mid \exists \boldsymbol{x} \in \mathbb{Z}_p^t \text{ s.t. } \boldsymbol{v} = g^{\boldsymbol{x}\cdot\mathbf{A}}\}$, the Jutla-Roy proofs only take $O(n-t)$ group elements – instead of $\Theta(n+t)$ in [33] – at the expense of settling for computational soundness. While highly efficient in the case $t \approx n$, these proofs remain of linear size in n and may result in long proofs when $t \ll n$, as is the case in, *e.g.*, certain applications of the Naor-Yung paradigm [15]. In the general case, we are still lacking a method for building proofs of size $O(t)$ – at least without relying on non-falsifiable assumptions [45] – which contrasts with the situation in the random oracle model.

The problem is even harder if we aim for simulation-soundness. While the Jutla-Roy solutions [38] nicely interact with their one-time simulation-sound proofs [37], they do not seem to readily extend into unbounded simulation-sound (USS) proofs (where the adversary can see an arbitrary number of simulated proofs before outputting a proof of its own) while retaining the same efficiency. For this reason, although they can be applied in specific cases like [15], we cannot always use them in a modular way to build CCA2-secure encryption schemes in scenarios where security definitions involve many challenge ciphertexts.

Our Contributions. Recently, in [43], it was pointed out that structure-preserving signatures (SPS) [3, 2] with (additive) homomorphic properties have unexpected applications in the design of non-malleable structure-preserving commitments. Here, we greatly extend their range of applications and demonstrate that they can surprisingly be used (albeit non-generically) in the design of strongly non-malleable primitives like simulation-sound proofs and chosen-ciphertext-secure cryptosystems.

Concretely, we describe unbounded simulation-sound QA-NIZK proofs of *constant-size* for linear subspaces. The length of a proof does not depend on the number of equations or the number of variables, but only on the underlying assumption. Like those of [38], our proofs are computationally sound under standard assumptions[5]. Somewhat surprisingly, they are even asymptotically shorter than random-oracle-based proofs derived from Σ-protocols.

Moreover, our construction provides *unbounded* simulation-soundness. Under the Decision Linear assumption [10], we obtain QA-NIZK arguments consisting of 15 group elements and a one-time signature with its verification key. As it turns out, it is also the first unbounded simulation-sound proof system that does not involve quadratic pairing product equations or a CCA2-secure encryption scheme. Efficiency comparisons (given in Appendix E) show that we only need 20 group elements per proof where the best USS extension [15] of Groth-Sahai costs $6t + 2n + 52$ group elements. Under the k-linear assumption, the proof length becomes $O(k^2)$ and thus avoids any dependency on the subspace dimension. Our proof system builds on the linearly homomorphic structure-preserving signatures of Libert, Peters, Joye and Yung [43], which allow signing vectors of group elements without knowing their discrete logarithms.

[5] Note that these results do not contradict the impossibility results of Gentry and Wichs [30] because, in the quasi-adaptive setting, the CRS may hide a trapdoor that allows recognizing elements of the language. The proof of [30] applies to reductions that cannot efficiently detect when the adversary breaks the soundness property.

For applications, like CCA2 security [46, 51], where only one-time simulation-soundness is needed, we further optimize our proof system and obtain a relatively simulation-sound QA-NIZK proof system, as defined in [37], with constant-size proofs. Under the DLIN assumption (resp. the k-linear assumption), we achieve relative simulation-soundness with only 4 (resp. $k+2$) group elements!

As the first application of USS proofs, we construct a chosen-ciphertext-secure keyed-homomorphic encryption scheme with threshold decryption. Keyed-homomorphic encryption is a primitive, suggested by Emura *et al.* [24], where homomorphic ciphertext manipulations are only possible to a party holding a devoted evaluation key SK_h which, by itself, does not enable decryption. The scheme should provide IND-CCA2 security when the evaluation key is unavailable to the adversary and remain IND-CCA1 secure when SK_h is exposed. Other approaches to reconcile homomorphism and non-malleability were taken in [47–49, 12, 18] but they inevitably satisfy weaker security notions than adaptive chosen-ciphertext security [50]. The results of [24] showed that CCA2-security does not rule out homomorphicity when the capability to compute over encrypted data is restricted.

Emura *et al.* [24] gave realizations of chosen-ciphertext-secure keyed-homomorphic schemes based on hash proof systems [21]. However, these do not readily enable threshold decryption – as would be desirable in voting protocols – since valid ciphertexts are not publicly recognizable, which makes it harder to prove CCA security in the threshold setting. Moreover, these solutions are not known to satisfy the strongest security definition of [24]. The reason is that this definition seemingly requires a form of unbounded simulation-soundness. Our QA-NIZK proofs fulfill this requirement and provide an efficient CCA2-secure threshold keyed-homomorphic system where ciphertexts are 65% shorter than in instantiations of the same high-level idea using previous simulation-sound proofs.

Using our relatively simulation-sound QA-NIZK proofs, we then build adaptively secure non-interactive threshold cryptosystems with CCA2 security and improved efficiency. The constructions of Libert and Yung [42] were improved by Escala *et al.* [25]. So far, the most efficient solution is obtained from the Jutla-Roy results [37, 38] via relatively sound proofs [37]. Using our relatively sound QA-NIZK proof system, we shorten ciphertexts by $\Theta(k)$ elements under the k-linear assumption.

Our Techniques. In our unbounded simulation-sound proofs, each QA-NIZK proof can be seen as a Groth-Sahai NIWI proof of knowledge of a one-time linearly homomorphic signature on the vector that allegedly belongs to the linear subspace. Here, the NIWI proof is generated for a Groth-Sahai CRS that depends on the verification key of a one-time signature (following an idea of Malkin *et al.* [44]), the private key of which is used to sign the entire proof so as to prevent re-randomizations. The reason why it provides unbounded simulation-soundness is that, with non-negligible probability, the CRS is perfectly hiding on all simulated proofs and extractable in the adversarially-generated fake proof. Hence, if the adversary manages to prove membership of a vector outside the linear subspace, the reduction is able to extract a homomorphic signature that it would not have been able to compute itself, thereby breaking the DLIN assumption. At a high level, the system can be seen as a two-tier proof system made of a non-malleable proof of knowledge of a malleable proof of membership.

In our optimized relatively-sound proofs, we adapt ideas of Jutla and Roy [37] and combine the one-time linearly homomorphic signature of [43] with a smooth-projective hash function [21].

Our threshold keyed-homomorphic cryptosystem combines a hash proof system and a publicly verifiable USS proof that the ciphertext is well-formed. The keyed-homomorphic property is achieved by using the simulation trapdoor of the proof system as an evaluation key SK_h, allowing the evaluator to generate proofs without knowing the witnesses. As implicitly done in [24] in the case of hash proof systems, the simulation trapdoor is thus used in the scheme and not only in the security proof.

2 Background and Definitions

2.1 Quasi-Adaptive NIZK Proofs

Quasi-Adaptive NIZK (QA-NIZK) proofs are NIZK proofs where the CRS is allowed to depend on the specific language for which proofs have to be generated. The CRS is divided into a fixed part Γ,

produced by an algorithm K_0, and a language-dependent part ψ. However, there should be a single simulator for the entire class of languages.

Let λ be a security parameter. For public parameters Γ produced by K_0, let $\mathcal{D}_\Gamma$ be a probability distribution over a collection of relations $\mathcal{R} = \{R_\rho\}$ parametrized by a string ρ with an associated language $\mathcal{L}_\rho = \{x \mid \exists w : R_\rho(x, w) = 1\}$.

We consider proof systems where the prover and the verifier both take a label lbl as additional input. For example, this label can be the message-carrying part of an Elgamal-like encryption. Formally, a tuple of algorithms $(\mathsf{K}_0, \mathsf{K}_1, \mathsf{P}, \mathsf{V})$ is a QA-NIZK proof system for $\mathcal{R}$ if there exists a PPT simulator $(\mathsf{S}_1, \mathsf{S}_2)$ such that, for any PPT adversaries $\mathcal{A}_1, \mathcal{A}_2$ and $\mathcal{A}_3$, we have the following properties:

Quasi-Adaptive Completeness:

$$\Pr[\Gamma \leftarrow \mathsf{K}_0(\lambda);\ \rho \leftarrow D_\Gamma;\ \psi \leftarrow \mathsf{K}_1(\Gamma, \rho);\\ (x, w, \mathsf{lbl}) \leftarrow \mathcal{A}_1(\Gamma, \psi, \rho);\ \pi \leftarrow \mathsf{P}(\psi, x, w, \mathsf{lbl}) : \mathsf{V}(\psi, x, \pi, \mathsf{lbl}) = 1 \ \text{ if } R_\rho(x, w) = 1] = 1.$$

Quasi-Adaptive Soundness:

$$\Pr[\Gamma \leftarrow \mathsf{K}_0(\lambda);\ \rho \leftarrow D_\Gamma;\ \psi \leftarrow \mathsf{K}_1(\Gamma, \rho);\ (x, \pi, \mathsf{lbl}) \leftarrow \mathcal{A}_2(\Gamma, \psi, \rho) :\\ \mathsf{V}(\psi, x, \pi, \mathsf{lbl}) = 1 \ \wedge\ \neg(\exists w : R_\rho(x, w) = 1)] \in \mathsf{negl}(\lambda).$$

Quasi-Adaptive Zero-Knowledge:

$$\Pr[\Gamma \leftarrow \mathsf{K}_0(\lambda);\ \rho \leftarrow D_\Gamma;\ \psi \leftarrow \mathsf{K}_1(\Gamma, \rho)\ :\ \mathcal{A}_3^{\mathsf{P}(\psi,.,.)}(\Gamma, \psi, \rho) = 1]\\ \approx \Pr[\Gamma \leftarrow \mathsf{K}_0(\lambda);\ \rho \leftarrow D_\Gamma;\ (\psi, \tau_{sim}) \leftarrow \mathsf{S}_1(\Gamma, \rho)\ :\ \mathcal{A}_3^{\mathsf{S}(\psi,\tau_{sim},.,.,.)}(\Gamma, \psi, \rho) = 1],$$

where

- $\mathsf{P}(\psi, ., ., .)$ emulates the actual prover. It takes as input (x, w) and lbl and outputs a proof π if $(x, w) \in R_\rho$. Otherwise, it outputs $\perp$.
- $\mathsf{S}(\psi, \tau_{sim}, ., ., .)$ is an oracle that takes as input (x, w) and lbl. It outputs a simulated proof $\mathsf{S}_2(\psi, \tau_{sim}, x, \mathsf{lbl})$ if $(x, w) \in R_\rho$ and $\perp$ if $(x, w) \notin R_\rho$.

We assume that the CRS ψ contains an encoding of ρ, which is thus available to V. The definition of Quasi-Adaptive Zero-Knowledge requires a single simulator for the entire family of relations $\mathcal{R}$.

2.2 Simulation-Soundness and Relative Soundness

It is often useful to have a property called *simulation-soundness*, which requires that the adversary be unable to prove false statements even after having seen simulated proofs for possibly false statements.

Unbounded Simulation-Soundness: For any PPT adversary $\mathcal{A}_4$, it holds that

$$\Pr[\Gamma \leftarrow \mathsf{K}_0(\lambda);\ \rho \leftarrow D_\Gamma;\ (\psi, \tau_{sim}) \leftarrow \mathsf{S}_1(\Gamma, \rho);\ (x, \pi, \mathsf{lbl}) \leftarrow \mathcal{A}_4^{\mathsf{S}_2(\psi,\tau_{sim},.,.)}(\Gamma, \psi, \rho)\ :\\ \mathsf{V}(\psi, x, \pi, \mathsf{lbl}) = 1 \ \wedge\ \neg(\exists w : R_\rho(x, w) = 1) \ \wedge\ (x, \pi, \mathsf{lbl}) \notin Q] \in \mathsf{negl}(\lambda),$$

where the adversary is allowed unbounded access to an oracle $\mathsf{S}_2(\psi, \tau, ., .)$ that takes as input statement-label pairs (x, lbl) (where x may be outside $\mathcal{L}_\rho$) and outputs simulated proofs $\pi \leftarrow \mathsf{S}_2(\psi, \tau_{sim}, x, \mathsf{lbl})$ before updating the set $Q = Q \cup \{(x, \pi, \mathsf{lbl})\}$, which is initially empty.

In the weaker notion of one-time simulation-soundness, only one query to the S_2 oracle is allowed.

In some applications, one may settle for a weaker notion, called *relative soundness* by Jutla and Roy [37], which allows for more efficient proofs, especially in the single-theorem case. Informally,

relatively sound proof systems involve both a public verifier *and* a private verification algorithm, which has access to a trapdoor. For hard languages, the two verifiers should almost always agree on any adversarially-created proof. Moreover, the private verifier should not accept a non-trivial proof for a false statement, even if the adversary has already seen proofs for false statements.

A labeled single-theorem relatively sound QA-NIZK proof system is comprised of a quasi-adaptive labeled proof system $(\mathsf{K}_0, \mathsf{K}_1, \mathsf{P}, \mathsf{V})$ along with an efficient private verifier W and an efficient simulator $(\mathsf{S}_1, \mathsf{S}_2)$. Moreover, the following properties should hold for any PPT adversaries $(\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \mathcal{A}_4)$.

Quasi Adaptive Relative Single-Theorem Zero-Knowledge:

$$\begin{aligned}\Pr[\Gamma \leftarrow \mathsf{K}_0(\lambda);\ \rho \leftarrow D_\Gamma;\ \psi \leftarrow \mathsf{K}_1(\Gamma,\rho);\ (x,w,\mathsf{lbl},s) \leftarrow \mathcal{A}_1^{\mathsf{V}(\psi,.,.)}(\Gamma,\psi,\rho);\\ \pi \leftarrow \mathsf{P}(\psi,\rho,x,w,\mathsf{lbl}) : \mathcal{A}_2^{\mathsf{V}(\psi,.,.)}(\pi,s)=1]\\ \approx \Pr[\Gamma \leftarrow \mathsf{K}_0(\lambda);\ \rho \leftarrow D_\Gamma;\ (\psi,\tau) \leftarrow \mathsf{S}_1(\Gamma,\rho);\ (x,w,\mathsf{lbl},s) \leftarrow \mathcal{A}_1^{\mathsf{W}(\psi,\tau,.,.)}(\Gamma,\psi,\rho);\\ \pi \leftarrow \mathsf{S}_2(\psi,\rho,\tau,x,\mathsf{lbl}) : \mathcal{A}_2^{\mathsf{W}(\psi,\tau,.,.)}(\pi,s)=1],\end{aligned}$$

Here, $\mathcal{A}_1$ is restricted to choosing (x, w) such that $R_\rho(x, w) = 1$.

Quasi Adaptive Relative Single-Theorem Simulation-Soundness:

$$\begin{aligned}\Pr[\Gamma \leftarrow \mathsf{K}_0(\lambda);\ \rho \leftarrow D_\Gamma;\ (\psi,\tau) \leftarrow \mathsf{S}_1(\Gamma,\rho);\ (x,\mathsf{lbl},s) \leftarrow \mathcal{A}_3^{\mathsf{W}(\psi,\tau,.,.)}(\Gamma,\psi,\rho);\\ \pi \leftarrow \mathsf{S}_2(\psi,\rho,\tau,x,\mathsf{lbl}) : (x',\mathsf{lbl}',\pi') \leftarrow \mathcal{A}_4^{\mathsf{W}(\psi,\tau,.,.)}(s,\pi) :\\ (x,\pi,\mathsf{lbl}) \neq (x',\pi',\mathsf{lbl}')\ \wedge\ \nexists w' \text{ s.t. } R_\rho(x',w')=1\ \wedge\ \mathsf{W}(\psi,\tau,x',\mathsf{lbl}',\pi')=1] \in \mathsf{negl}(\lambda)\end{aligned}$$

Note that the definition of relative simulation-soundness does not require the adversary to provide a witness but the definition of single-theorem zero-knowledge does.

2.3 Definitions for Threshold Keyed-Homomorphic Encryption

A (t, N)-threshold keyed-homomorphic encryption scheme consists of the following algorithms.

Keygen(λ, t, N)**:** takes as input a security parameter λ and integers $t, N \in \mathsf{poly}(\lambda)$ (with $1 \leq t \leq N$), where N is the number of decryption servers and $t \leq N$ is the decryption threshold. It outputs $(PK, SK_h, \mathbf{VK}, \mathbf{SK}_d)$, where PK is the public key, SK_h is the homomorphic evaluation key, $\mathbf{SK}_d = (SK_{d,1}, \ldots, SK_{d,N})$ is a vector of private key shares and $\mathbf{VK} = (VK_1, \ldots, VK_N)$ is a vector of verification keys. For each i, the decryption server i is given the share $(i, SK_{d,i})$. The verification key VK_i will be used to check the validity of decryption shares generated using $SK_{d,i}$.

Encrypt(PK, M)**:** takes a input a public key PK and a plaintext M. It outputs a ciphertext C.

Ciphertext-Verify(PK, C)**:** takes as input a public key PK and a ciphertext C. It outputs 1 if C is deemed valid w.r.t. PK and 0 otherwise.

Share-Decrypt$(PK, i, SK_{d,i}, C)$**:** on input of a public key PK, a ciphertext C and a private-key share $(i, SK_{d,i})$, this (possibly randomized) algorithm outputs a special symbol $(i, \perp)$ if **Ciphertext-Verify**$(PK, C) = 0$. Otherwise, it outputs a decryption share $\mu_i = (i, \hat{\mu}_i)$.

Share-Verify(PK, VK_i, C, μ_i)**:** takes in PK, the verification key VK_i, a ciphertext C and a purported decryption share $\mu_i = (i, \hat{\mu}_i)$. It outputs either 1 or 0. In the former case, μ_i is said to be a *valid* decryption share. We adopt the convention that $(i, \perp)$ is an invalid decryption share.

Combine$(PK, \mathbf{VK}, C, \{\mu_i\}_{i\in S})$**:** takes in $(PK, \mathbf{VK}, C)$ and a t-subset $S \subset \{1, \ldots, N\}$ with decryption shares $\{\mu_i\}_{i\in S}$. It outputs either a plaintext M or $\perp$ if $\{\mu_i\}_{i\in S}$ contains invalid shares.

Eval$(PK, SK_h, C^{(1)}, C^{(2)})$**:** takes as input PK, the evaluation key SK_h and ciphertexts $C^{(1)}$, $C^{(2)}$. If **Ciphertext-Verify**$(PK, C^{(j)}) = 0$ for some $j \in \{1, 2\}$, the algorithm returns $\perp$. Otherwise, it conducts a binary homomorphic operation over $C^{(1)}$ and $C^{(2)}$ and outputs a ciphertext C.

The above syntax assumes a trusted dealer. It generalizes that of ordinary threshold cryptosystems. By setting $SK_h = \varepsilon$ and discarding the evaluation algorithm, we obtain a classical threshold system.

Definition 1. *A threshold keyed-homomorphic public-key cryptosystem is secure against chosen-ciphertext attacks (or KH-CCA secure) if no PPT adversary has noticeable advantage in this game:*

1. *The challenger runs* $\mathbf{Keygen}(\lambda)$ *to obtain a public key* PK*, vectors* $\mathbf{SK}_d = (SK_{d,1}, \ldots, SK_{d,N})$*,* $\mathbf{VK} = (VK_1, \ldots, VK_N)$ *and a homomorphic evaluation key* SK_h*. It gives* PK *and* $\mathbf{VK}$ *to the adversary* $\mathcal{A}$ *and keeps* $(SK_h, \mathbf{SK}_d)$ *to itself. In addition, the challenge initializes a set* $\mathcal{D} \leftarrow \emptyset$*, which is initially empty.*
2. *The adversary* $\mathcal{A}$ *adaptively makes queries to the following oracles on multiple occasions:*
   - *Corruption query: at any time,* $\mathcal{A}$ *may decide to corrupt a decryption server. To this end, it specifies an index* $i \in \{1, \ldots, N\}$ *and obtains the private key share* $SK_{d,i}$*.*
   - *Evaluation query:* $\mathcal{A}$ *can invoke the evaluation oracle* $\mathsf{Eval}(SK_h, .)$ *on a pair* $(C^{(1)}, C^{(2)})$ *of ciphertexts of its choice. If there exists* $j \in \{1, 2\}$ *such that* $\mathbf{Ciphertext\text{-}Verify}(PK, C^{(j)}) = 0$*, return* $\perp$*. Otherwise, the oracle* $\mathsf{Eval}(SK_h, .)$ *computes* $C \leftarrow \mathbf{Eval}(SK_h, C^{(1)}, C^{(2)})$ *and returns* C*. In addition, if* $C^{(1)} \in \mathcal{D}$ *or* $C^{(2)} \in \mathcal{D}$*, it sets* $\mathcal{D} \leftarrow \mathcal{D} \cup \{C\}$*.*
   - *Reveal query: at any time,* $\mathcal{A}$ *may also decide to corrupt the evaluator by invoking the* RevHK *oracle on a unique occasion. The oracle responds by returning* SK_h*.*
   - *Decryption query:* $\mathcal{A}$ *can also invoke the partial decryption oracle on arbitrary ciphertexts* C *and indexes* $i \in \{1, \ldots, n\}$*. If* $\mathbf{Ciphertext\text{-}Verify}(PK, C) = 0$ *or if* $C \in \mathcal{D}$*, the oracle returns* $\perp$*. Otherwise, the oracle returns the decryption share* $\mu_i \leftarrow \mathbf{Share\text{-}Decrypt}(PK, i, SK_{d,i}, C)$*.*
3. *The adversary* $\mathcal{A}$ *chooses two equal-length messages* M_0, M_1 *and obtains* $C^\star = \mathbf{Encrypt}(PK, M_\beta)$ *for some random bit* $\beta \xleftarrow{R} \{0, 1\}$*. In addition, the challenger sets* $\mathcal{D} \leftarrow \mathcal{D} \cup \{C^\star\}$*.*
4. $\mathcal{A}$ *makes further queries as in step 2 with some restrictions. Namely,* $\mathcal{A}$ *cannot corrupt more than* $t - 1$ *servers throughout the entire game. Moreover, if* $\mathcal{A}$ *chooses to obtain* SK_h *(via the* RevHK *oracle) at some point, no more post-challenge decryption query is allowed beyond that point.*
5. $\mathcal{A}$ *outputs a bit* β' *and is deemed successful if* $\beta' = \beta$*. As usual,* $\mathcal{A}$*'s advantage is measured as the distance* $\mathbf{Adv}(\mathcal{A}) = |\Pr[\beta' = \beta] - \frac{1}{2}|$*.*

It is important to note that, even if $\mathcal{A}$ chooses to obtain SK_h immediately after having seen the public key PK, it still has access to the decryption oracle *before* the challenge phase. In other words, the scheme should remain IND-CCA1 if $\mathcal{A}$ is given PK and SK_h at the outset of the game. After the challenge phase, decryption queries are allowed until the moment when the adversary obtains SK_h.

In [24], Emura *et al.* suggested a weaker definition where the adversary is not allowed to query the evaluation oracle on derivatives of the challenge ciphertext. As a consequence, the set $\mathcal{D}$ is always the singleton $\{C^\star\}$ after step 3. In this paper, we will stick to the stronger definition.

2.4 Hardness Assumptions

For simplicity, we use symmetric bilinear maps $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ over groups of prime order p, but extensions to the asymmetric setting $e : \mathbb{G} \times \hat{\mathbb{G}} \to \mathbb{G}_T$ are possible.

Definition 2 ([10]). *The* **Decision Linear Problem** *(DLIN) in* $\mathbb{G}$*, is to distinguish the distributions* $(g^a, g^b, g^{ac}, g^{bd}, g^{c+d})$ *and* $(g^a, g^b, g^{ac}, g^{bd}, g^z)$*, with* $a, b, c, d \xleftarrow{R} \mathbb{Z}_p$*,* $z \xleftarrow{R} \mathbb{Z}_p$*.*

We sometimes use the Simultaneous Double Pairing (SDP) assumption, which is weaker than DLIN. As noted in [17], any algorithm solving SDP immediately yields a DLIN distinguisher.

Definition 3. *The* **Simultaneous Double Pairing problem** *(SDP) in* $(\mathbb{G}, \mathbb{G}_T)$ *is, given group elements* $(g_z, g_r, h_z, h_u) \in \mathbb{G}^4$*, to find a non-trivial triple* $(z, r, u) \in \mathbb{G}^3 \backslash \{(1_\mathbb{G}, 1_\mathbb{G}, 1_\mathbb{G})\}$ *such that* $e(g_z, z) \cdot e(g_r, r) = 1_{\mathbb{G}_T}$ *and* $e(h_z, z) \cdot e(h_u, u) = 1_{\mathbb{G}_T}$*.*

2.5 Linearly Homomorphic Structure-Preserving Signatures

Linearly homomorphic SPS schemes are homomorphic signatures where messages and signatures live in the domain group $\mathbb{G}$ (see Appendix B for syntactic definitions) of a bilinear map. Libert *et al.* [43] described the following one-time construction and proved its security under the SDP assumption. By "one-time", we mean that only one linear subspace can be safely signed using a given key pair.

Keygen(λ, n): given a security parameter λ and the dimension $n \in \mathbb{N}$ of vectors to be signed, choose bilinear group $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$. Choose $g_z, g_r, h_z, h_u \xleftarrow{R} \mathbb{G}$. Then, for $i = 1$ to n, pick $\chi_i, \gamma_i, \delta_i \xleftarrow{R} \mathbb{Z}_p$ and compute $g_i = g_z^{\chi_i} g_r^{\gamma_i}$ and $h_i = h_z^{\chi_i} h_u^{\delta_i}$. The private key is $\mathsf{sk} = \{(\chi_i, \gamma_i, \delta_i)\}_{i=1}^n$ while the public key is $\mathsf{pk} = \big(g_z, g_r, h_z, h_u, \{(g_i, h_i)\}_{i=1}^n\big)$.

Sign($\mathsf{sk}, (M_1, \ldots, M_n)$): to sign a vector $(M_1, \ldots, M_n) \in \mathbb{G}^n$ using $\mathsf{sk} = \{(\chi_i, \gamma_i, \delta_i)\}_{i=1}^n$, compute and return $(z, r, u) = \big(\prod_{i=1}^n M_i^{-\chi_i}, \ \prod_{i=1}^n M_i^{-\gamma_i}, \ \prod_{i=1}^n M_i^{-\delta_i}\big) \in \mathbb{G}^3$.

SignDerive($\mathsf{pk}, \{(\omega_i, \sigma^{(i)})\}_{i=1}^\ell$): given a public key pk and ℓ tuples $(\omega_i, \sigma^{(i)})$, where $\omega_i \in \mathbb{Z}_p$ for each i, parse $\sigma^{(i)}$ as $\sigma^{(i)} = \big(z_i, r_i, u_i\big) \in \mathbb{G}^3$ for $i = 1$ to ℓ. Then, compute and return $\sigma = (z, r, u)$, where $z = \prod_{i=1}^\ell z_i^{\omega_i}$, $r = \prod_{i=1}^\ell r_i^{\omega_i}$ and $u = \prod_{i=1}^\ell u_i^{\omega_i}$.

Verify($\mathsf{pk}, \sigma, (M_1, \ldots, M_n)$): given a signature $\sigma = (z, r, u) \in \mathbb{G}^3$ and a vector $(M_1, \ldots, M_n)$, return 1 if and only if $(M_1, \ldots, M_n) \neq (1_\mathbb{G}, \ldots, 1_\mathbb{G})$ and (z, r, u) satisfy

$$1_{\mathbb{G}_T} = e(g_z, z) \cdot e(g_r, r) \cdot \prod_{i=1}^n e(g_i, M_i), \qquad 1_{\mathbb{G}_T} = e(h_z, z) \cdot e(h_u, u) \cdot \prod_{i=1}^n e(h_i, M_i). \tag{1}$$

One particularity of this scheme is that, even if the private key is available, it is difficult to find two distinct signatures on the same vector if the SDP assumption holds: by dividing out the two signatures, one obtains the solution of an SDP instance (g_z, g_r, h_z, h_u) contained in the public key.

Two constructions of full-fledged (as opposed to one-time) linearly homomorphic SPS were given in [43]. One of these will serve as a basis for our proof system and is recalled in Appendix C. In these constructions, all algorithms additionally input a tag which identifies the dataset that vectors belongs to. Importantly, only vectors associated with the same tag can be homomorphically combined.

3 Unbounded Simulation-Sound Quasi-Adaptive NIZK Arguments

In the following, vectors are always considered as row vectors unless stated otherwise. If $\mathbf{A} \in \mathbb{Z}_p^{t \times n}$ is a matrix, we denote by $g^\mathbf{A} \in \mathbb{G}^{t \times n}$ the matrix obtained by exponentiating g using the entries of $\mathbf{A}$.

We consider public parameters $\Gamma = (\mathbb{G}, \mathbb{G}_T, g)$ consisting of bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ with a generator $g \in \mathbb{G}$. Like [38], we will consider languages $\mathcal{L}_\rho = \{g^{\boldsymbol{x} \cdot \mathbf{A}} \in \mathbb{G}^n \mid \boldsymbol{x} \in \mathbb{Z}_p^t\}$ that are parametrized by $\rho = g^\mathbf{A} \in \mathbb{G}^{t \times n}$, where $\mathbf{A} \in \mathbb{Z}_q^{t \times n}$ is a $t \times n$ matrix of rank $t < n$.

As in [38], we assume that the distribution $\mathcal{D}_\Gamma$ is efficiently samplable: there exists a PPT algorithm which outputs a pair $(\rho, \mathbf{A})$ describing a relation R_ρ and its associated language $\mathcal{L}_\rho$ according to $\mathcal{D}_\Gamma$. One example of such a distribution is obtained by picking a uniform matrix $\mathbf{A} \xleftarrow{R} \mathbb{Z}_p^{t \times n}$ – which has full rank with overwhelming probability – and setting $\rho = g^\mathbf{A}$.

Our construction builds on the homomorphic signature recalled in Appendix C. Specifically, the language-dependent CRS ψ contains one-time linearly homomorphic signatures on the rows of the matrix $\rho \in \mathbb{G}^{t \times n}$. For each vector $\boldsymbol{v} \in \mathcal{L}_\rho$, the prover can use the witness $\boldsymbol{x} \in \mathbb{Z}_p^t$ to derive and prove knowledge of a one-time homomorphic signature (z, r, u) on $\boldsymbol{v}$. This signature (z, r, u) is already a QA-NIZK proof of membership but it does not provide simulation-soundness. To acquire this property, we follow [44] and generate a NIWI proof of knowledge of (z, r, u) for a Groth-Sahai CRS that depends on the verification key of an ordinary one-time signature. The latter's private key is used to sign the NIWI proof so as to prevent unwanted proof manipulations. Using the private key of the homomorphic one-time signature as a trapdoor, the simulator is also able to create proofs for vectors

$\boldsymbol{v} \notin \mathcal{L}_\rho$. Due to the use of perfectly NIWI proofs, these fake proofs do not leak any more information about the simulation key than the CRS does. At the same time, the CRS can be prepared in such a way that, with non-negligible probability, it becomes perfectly binding on an adversarially-generated proof, which allows extracting a non-trivial signature on a vector $\boldsymbol{v} \notin \mathcal{L}_\rho$.

Like [38], our quasi-adaptive NIZK proof system $(\mathsf{K}_0, \mathsf{K}_1, \mathsf{P}, \mathsf{V})$ is a split CRS construction in that K_1 can be divided into two algorithms $(\mathsf{K}_{10}, \mathsf{K}_{11})$. The first one K_{10} outputs some state information s and a first CRS $\mathbf{CRS}_2$ which is only used by the verifier and does not depend on the language $\mathcal{L}_\rho$. The second part K_{11} of K_1 inputs the state information s and the output of Γ of K_0 and outputs $\mathbf{CRS}_1$ which is only used by the prover. The construction goes as follows.

$\mathsf{K}_0(\lambda)$**:** choose groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$ with $g \xleftarrow{R} \mathbb{G}$. Then, output $\Gamma = (\mathbb{G}, \mathbb{G}_T, g)$

The dimensions (t, n) of the matrix $\mathbf{A} \in \mathbb{Z}_p^{t \times n}$ can be either fixed or part of the language, so that t, n can be given as input to the CRS generation algorithm K_1.

$\mathsf{K}_1(\Gamma, \rho)$**:** parse Γ as $(\mathbb{G}, \mathbb{G}_T, g)$ and ρ as a matrix $\rho = \big(G_{i,j}\big)_{1 \leq i \leq t,\ 1 \leq j \leq n} \in \mathbb{G}^{t \times n}$.

1. Generate a key pair $(\mathsf{pk}_{rand}, \mathsf{sk}_{rand})$ for the randomizable signature of Appendix C in order to sign vectors of $\mathbb{G}^n$. Namely, choose $g_z, g_r, h_z, h_u \xleftarrow{R} \mathbb{G}$ and do the following.
   a. For $i = 1$ to n, pick $\chi_i, \gamma_i, \delta_i \xleftarrow{R} \mathbb{Z}_p$ and compute $g_i = g_z{}^{\chi_i} g_r{}^{\gamma_i}$ and $h_i = h_z{}^{\chi_i} h_u{}^{\delta_i}$.
   b. Generate $L + 1$ Groth-Sahai common reference strings, for some $L \in \mathsf{poly}(\lambda)$. To this end, choose $f_1, f_2 \xleftarrow{R} \mathbb{G}$ and define $\boldsymbol{f}_1 = (f_1, 1, g) \in \mathbb{G}^3$, $\boldsymbol{f}_2 = (1, f_2, g) \in \mathbb{G}^3$. Then, pick $\boldsymbol{f}_{3,i} \xleftarrow{R} \mathbb{G}^3$ for $i = 0$ to L.

   Let $\mathsf{sk}_{rand} = \{(\chi_i, \gamma_i, \delta_i)\}_{i=1}^n$ be the private key and the matching public key is

$$\mathsf{pk}_{rand} = \Big(g_z,\ g_r,\ h_z,\ h_u,\ \{(g_i, h_i)\}_{i=1}^n,\ \mathbf{f} = \big(\boldsymbol{f}_1, \boldsymbol{f}_2, \{\boldsymbol{f}_{3,i}\}_{i=0}^L\big) \Big).$$

2. Use sk_{rand} to generate one-time linearly homomorphic signatures $\{(z_i, r_i, u_i)\}_{i=1}^t$ on the vectors $\boldsymbol{\rho}_i = (G_{i1}, \ldots, G_{in}) \in \mathbb{G}^n$ that form the rows of ρ. These are obtained as

$$(z_i, r_i, u_i) = \big(\prod_{j=1}^n G_{i,j}^{-\chi_j},\ \prod_{j=1}^n G_{i,j}^{-\gamma_j},\ \prod_{j=1}^n G_{i,j}^{-\delta_j}\big) \qquad \forall i \in \{1, \ldots, t\}.$$

3. Choose a strongly unforgeable one-time signature $\Sigma = (\mathcal{G}, \mathcal{S}, \mathcal{V})$ with verification keys consisting of L-bit strings.
4. The CRS $\psi = (\mathbf{CRS}_1, \mathbf{CRS}_2)$ consists of two parts which are defined as

$$\mathbf{CRS}_1 = \Big(\rho,\ \mathsf{pk}_{rand},\ \{(z_i, r_i, u_i)\}_{i=1}^t,\ \Sigma\Big), \qquad \mathbf{CRS}_2 = \Big(\mathsf{pk}_{rand},\ \Sigma\Big),$$

   while the simulation trapdoor τ_{sim} is $\mathsf{sk}_{rand} = \{(\chi_i, \gamma_i, \delta_i)\}_{i=1}^n$.

$\mathsf{P}(\Gamma, \psi, \boldsymbol{v}, x, \mathsf{lbl})$**:** given a candidate $\boldsymbol{v} \in \mathbb{G}^n$ and a witness $\boldsymbol{x} = (x_1, \ldots, x_t) \in \mathbb{Z}_p^t$ such that $\boldsymbol{v} = g^{\boldsymbol{x} \cdot \mathbf{A}}$, generate a one-time signature key pair $(\mathsf{SVK}, \mathsf{SSK}) \leftarrow \mathcal{G}(\lambda)$ and do the following.

1. Using $\{(z_j, r_j, u_j)\}_{j=1}^t$, derive a one-time linearly homomorphic signature (z, r, u) on $\boldsymbol{v}$. Namely, compute $z = \prod_{i=1}^t z_i^{x_i}$, $r = \prod_{i=1}^t r_i^{x_i}$ and $u = \prod_{i=1}^t u_i^{x_i}$.
2. Using $\mathsf{SVK} = \mathsf{SVK}[1] \ldots \mathsf{SVK}[L] \in \{0,1\}^L$, define the vector $\boldsymbol{f}_{\mathsf{SVK}} = \boldsymbol{f}_{3,0} \cdot \prod_{i=1}^L \boldsymbol{f}_{3,i}^{\mathsf{SVK}[i]}$ and assemble a Groth-Sahai CRS $\mathbf{f}_{\mathsf{SVK}} = (\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{f}_{\mathsf{SVK}})$. Using $\mathbf{f}_{\mathsf{SVK}}$, generate commitments $\boldsymbol{C}_z$, $\boldsymbol{C}_r$, $\boldsymbol{C}_u$ to the components of $(z, r, u) \in \mathbb{G}^3$ along with NIWI proofs $(\boldsymbol{\pi}_1, \boldsymbol{\pi}_2)$ that $\boldsymbol{v}$ and (z, r, u) satisfy (1). Let $(\boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2) \in \mathbb{G}^{15}$ be the resulting commitments and proofs.
3. Generate $\sigma = \mathcal{S}(\mathsf{SSK}, (\boldsymbol{v}, \boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \mathsf{lbl}))$ and output

$$\pi = (\mathsf{SVK}, \boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \sigma) \tag{2}$$

$\mathsf{V}(\Gamma, \psi, \boldsymbol{v}, \pi, \mathsf{lbl})$**:** parse π as per (2) and return 1 if (i) $\mathcal{V}(\mathsf{SVK}, (\boldsymbol{v}, \boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \mathsf{lbl}), \sigma) = 1$; (ii) $(\boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2)$ forms a valid NIWI proof for the CRS $\mathbf{f}_{\mathsf{SVK}} = (\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{f}_{\mathsf{SVK}})$ (see (4) in Appendix C for the detailed verification equations). If either condition fails to hold, return 0.

In order to simulate a proof for a given vector $\boldsymbol{v} \in \mathbb{G}^n$, the simulator uses $\tau_{sim} = \mathsf{sk}_{rand}$ to generate a fresh one-time homomorphic signature on $\boldsymbol{v} \in \mathbb{G}^n$ and proceeds as in steps 2-3 of algorithm P.

The proof π only consists of 15 group elements and a one-time pair (SVK, σ). Remarkably, its length does not depend on the number of equations n or the number of variables t. In comparison, Groth-Sahai proofs already require $3t + 2n$ group elements in their basic form and become even more expensive when it comes to achieve unbounded simulation-soundness. The Jutla-Roy techniques [38] reduce the proof length to $2(n - t)$ elements – which only competes with our proofs when $t \approx n$ – but it is unclear how to extend them to get unbounded simulation-soundness without affecting their efficiency. Our CRS consists of $O(t + n + L)$ group elements against $O(t(n - t))$ in [38]. More detailed comparisons are given in Appendix E between proof systems based on the DLIN assumption.

Interestingly, the above scheme even outperforms Fiat-Shamir-like proofs derived from Σ-protocols which would give $\Theta(t)$-size proofs here. The construction readily extends to rely on the k-linear assumption for $k > 2$. In this case, the proof comprises $(k + 1)(2k + 1)$ elements and its size thus only depends on k, as detailed in Appendix D.

Moreover, the verification algorithm only involves *linear* pairing product equations whereas all known unbounded simulation-sound extensions of Groth-Sahai proofs require either quadratic equations or a linearization step involving extra variables.

We finally remark that, if we give up the simulation-soundness property, the proof length drops to $k + 1$ group elements under the k-linear assumption.

Theorem 1. *The scheme is an unbounded simulation-sound QA-NIZK proof system if the DLIN assumption holds in $\mathbb{G}$ and Σ is strongly unforgeable.* (The proof is given in Appendix F).

We note that the above construction is not tightly secure as the gap between the simulation-soundness adversary's advantage and the probability to break the DLIN assumption depends on the number of simulated proofs obtained by the adversary. For applications like tightly secure public-key encryption [34], it would be interesting to modify the proof system to obtain tight security.

4 Single-Theorem Relatively Sound Quasi-Adaptive NIZK Arguments

In applications where single-theorem relatively sound NIZK proofs suffice, we can further improve the efficiency. Under the k-linear assumption, the proof length reduces from $O(k^2)$ elements to $O(k)$ elements. Under the DLIN assumption, each proof fits within 4 elements and only costs $2n+6$ pairings to verify. In comparison, the verifier needs $2(n - t)(t + 2)$ pairing evaluations in [38].

As in [37], we achieve relative soundness using smooth projective hash functions [21]. To this end, we need to encode the matrix $\rho \in \mathbb{G}^{t \times n}$ as a $2t \times (2n + 1)$ matrix.

$\mathsf{K}_0(\lambda)$**:** choose groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$ with $g \xleftarrow{R} \mathbb{G}$. Then, output $\Gamma = (\mathbb{G}, \mathbb{G}_T, g)$.

Again, the dimensions of $\mathbf{A} \in \mathbb{Z}_p^{t \times n}$ can be either fixed or part of $\mathcal{L}_\rho$, so that t, n can be given as input to the CRS generation algorithm K_1.

$\mathsf{K}_1(\Gamma, \rho)$**:** parse Γ as $(\mathbb{G}, \mathbb{G}_T, g)$ and ρ as $\rho = \left(G_{ij}\right)_{1 \le i \le t,\ 1 \le j \le n} \in \mathbb{G}^{t \times n}$ and do the following.

1. Choose two n-vectors $\boldsymbol{d} = (d_1, \ldots, d_n) \xleftarrow{R} \mathbb{Z}_p^n$ and $\boldsymbol{e} = (e_1, \ldots, e_n) \xleftarrow{R} \mathbb{Z}_p^n$ in order to define $\boldsymbol{W} = (W_1, \ldots, W_t) = g^{\mathbf{A} \cdot \boldsymbol{d}^\top} \in \mathbb{G}^t$ and $\boldsymbol{Y} = (Y_1, \ldots, Y_t) = g^{\mathbf{A} \cdot \boldsymbol{e}^\top} \in \mathbb{G}^t$. These will be used to define a projective hash function.

2. Generate a key pair $(\mathsf{pk}_{ots}, \mathsf{sk}_{ots})$ for the one-time linearly homomorphic signature of Section 2.5 in order to sign vectors in $\mathbb{G}^{2n+1}$. Let $\mathsf{pk}_{ots} = \big((\mathbb{G}, \mathbb{G}_T), g_z, g_r, h_z, h_u, \{(g_i, h_i)\}_{i=1}^{2n+1}\big)$ be the public key and let $\mathsf{sk}_{ots} = \{(\chi_i, \gamma_i, \delta_i)\}_{i=1}^{2n+1}$ be the corresponding private key.
3. Use sk_{ots} to generate one-time linearly homomorphic signatures $\{(z_i, r_i, u_i)\}_{i=1}^{2t}$ on the independent vectors below, which are obtained from the rows of the matrix $\rho = \big(G_{i,j}\big)_{1 \le i \le t,\ 1 \le j \le n}$.

$$\begin{aligned} \boldsymbol{H}_{2i-1} &= (G_{i,1}, \ldots, G_{i,n}, Y_i,\ 1\ ,\ \ \ldots\ \ ,1\ ) \in \mathbb{G}^{2n+1} \qquad\qquad i \in \{1, \ldots, t\} \\ \boldsymbol{H}_{2i} &= (1\ ,\ \ \ldots\ \ ,1\ , W_i, G_{i,1}, \ldots, G_{i,n}) \in \mathbb{G}^{2n+1} \end{aligned}$$

4. Choose a collision-resistant hash function $H : \{0,1\}^* \to \mathbb{Z}_p$.
5. The CRS ψ consists of a first part $\mathbf{CRS}_1$ that is only used by the prover and a second part $\mathbf{CRS}_2$ which is only used by the verifier. These are defined as

$$\mathbf{CRS}_1 = \Big(\rho,\ \mathsf{pk}_{ots},\ \boldsymbol{W},\ \boldsymbol{Y},\ \{(z_i, r_i, u_i)\}_{i=1}^{2t},\ H\Big), \qquad \mathbf{CRS}_2 = \Big(\mathsf{pk}_{ots},\ \boldsymbol{W},\ \boldsymbol{Y},\ H\Big).$$

The simulation trapdoor τ_{sim} is sk_{ots} and the private verification trapdoor is $\tau_v = \{\boldsymbol{d}, \boldsymbol{e}\}$.

$\mathsf{P}(\Gamma, \psi, \boldsymbol{v}, x, \mathsf{lbl})$**:** given a candidate $\boldsymbol{v} \in \mathbb{G}^n$, a witness $\boldsymbol{x} = (x_1, \ldots, x_t) \in \mathbb{Z}_p^t$ such that $\boldsymbol{v} = g^{\boldsymbol{x} \cdot \mathbf{A}}$ and a label lbl, compute $\alpha = H(\rho, \boldsymbol{v}, \mathsf{lbl}) \in \mathbb{Z}_p$. Then, using $\{(z_i, r_i, u_i)\}_{i=1}^{2t}$, derive a one-time linearly homomorphic signature (z, r, u) on the vector $\tilde{\boldsymbol{v}} = \big(v_1, \ldots, v_n, \pi_0, v_1^\alpha, \ldots, v_n^\alpha\big) \in \mathbb{G}^{2n+1}$, where $\pi_0 = \prod_{i=1}^t (W_i^\alpha Y_i)^{x_i}$. Namely, compute and output the proof $\pi = (z, r, u, \pi_0) \in \mathbb{G}^4$, where

$$z = \prod_{i=1}^t (z_{2i-1} \cdot z_{2i}^\alpha)^{x_i}, \qquad r = \prod_{i=1}^t (r_{2i-1} \cdot r_{2i}^\alpha)^{x_i}, \qquad u = \prod_{i=1}^t (u_{2i-1} \cdot u_{2i}^\alpha)^{x_i} \qquad \pi_0 = \prod_{i=1}^t (W_i^\alpha Y_i)^{x_i}$$

$\mathsf{V}(\Gamma, \psi, \boldsymbol{v}, \pi, \mathsf{lbl})$**:** parse $\boldsymbol{v}$ as $(v_1, \ldots, v_n) \in \mathbb{G}^n$ and π as $(z, r, u, \pi_0) \in \mathbb{G}^4$. Compute $\alpha = H(\rho, \boldsymbol{v}, \mathsf{lbl})$ and return 1 if and only if (z, r, u) is a valid signature on $\tilde{\boldsymbol{v}} = (v_1, \ldots, v_n, \pi_0, v_1^\alpha, \ldots, v_n^\alpha) \in \mathbb{G}^{2n+1}$. Namely, it should satisfy the equalities $1_{\mathbb{G}_T} = e(g_z, z) \cdot e(g_r, r) \cdot \prod_{i=1}^n e(g_i \cdot g_{i+n+1}^\alpha, v_i) \cdot e(g_{n+1}, \pi_0)$ and $1_{\mathbb{G}_T} = e(h_z, z) \cdot e(h_u, u) \cdot \prod_{i=1}^n e(h_i \cdot h_{i+n+1}^\alpha, v_i) \cdot e(h_{n+1}, \pi_0)$.

$\mathsf{W}(\Gamma, \psi, \tau_v, \boldsymbol{v}, \pi, \mathsf{lbl})$**:** given $\boldsymbol{v} = (v_1, \ldots, v_n) \in \mathbb{G}^n$, parse π as $(z, r, u, \pi_0) \in \mathbb{G}^4$ and τ_v as $\{\boldsymbol{d}, \boldsymbol{e}\}$, with $\boldsymbol{d} = (d_1, \ldots, d_n) \in \mathbb{Z}_p^n$ and $\boldsymbol{e} = (e_1, \ldots, e_n) \in \mathbb{Z}_p^n$. Compute $\alpha = H(\rho, \boldsymbol{v}, \mathsf{lbl}) \in \mathbb{Z}_p$ and return 0 if the public verification test V fails. Otherwise, return 1 if $\pi_0 = \prod_{j=1}^n v_j^{e_j + \alpha d_j}$ and 0 otherwise.

We note that, while the proving algorithm is deterministic, each statement has many valid proofs. However, finding two valid proofs for the same statement is computationally hard, as will be shown in the proof of Theorem 2.

The scheme readily extends to rest on the k-linear assumption with $k > 2$. In this case, the proof requires $k + 2$ group elements – whereas combining the techniques of [37, 38] demands $k(n + 1 - t)$ elements per proof – and a CRS of size $O(k(n+t))$. From a security standpoint, we prove the following result in Appendix G.

Theorem 2. *The above proof system is a relatively sound QA-NIZK proof system if the SDP assumption holds in $(\mathbb{G}, \mathbb{G}_T)$ and if H is a collision-resistant hash function.*

As an application, we describe a new adaptively secure CCA2-secure non-interactive threshold cryptosystem based on the DLIN assumption in Appendix I. Under the k-linear assumption, the scheme provides ciphertexts that are $\Theta(k)$ group elements shorter than in previous such constructions. Under the DLIN assumption, ciphertexts consist of 8 elements of $\mathbb{G}$, which spares one group element w.r.t. the best previous variants [37, 38] of Cramer-Shoup with publicly verifiable ciphertexts.

5 An Efficient Threshold Keyed-Homomorphic KH-CCA-Secure Encryption Scheme from the DLIN Assumption

The use of linearly homomorphic signatures as publicly verifiable proofs of ciphertext validity in the Cramer-Shoup paradigm [20, 21] was suggested in [43]. However, the latter work only discusses non-adaptive (*i.e.*, CCA1) attacks. In the CCA2 case, a natural idea is to proceed as in our unbounded simulation-sound proof system and use the verification key of a on-time signature as the tag of a homomorphic signature: since cross-tag homomorphic operations are disallowed, the one-time signature will prevent illegal ciphertext manipulations after the challenge phase.

To obtain the desired keyed-homomorphic property, we use the simulation trapdoor of a simulation-sound proof system as the homomorphic evaluation key. This approach was already used by Emura *et al.* [24] in the context of designated verifier proofs. Here, publicly verifiable proofs are obtained from a homomorphic signature scheme of which the private key serves as an evaluation key: anyone equipped with this key can multiply two ciphertexts (or, more precisely, their built-in homomorphic components), generate a new tag and sign the resulting ciphertext using the private key of the homomorphic signature. Moreover, we can leverage the fact that the latter private key is always available to the reduction in the security proof of the homomorphic signature [43]. In the game of Definition 1, the simulator can thus hand over the evaluation key SK_h to the adversary upon request.

Emura *et al.* [24] gave constructions of KH-CCA secure encryption schemes based on hash proof systems [21]. However, these constructions are only known to provide a relaxed flavor of KH-CCA security where evaluation queries should not involve derivatives of the challenge ciphertext. The reason is that 2-universal hash proof systems [21] only provide a form of one-time simulation soundness whereas the model of Definition 1 seemingly requires unbounded simulation-soundness. Indeed, when the evaluation oracle is queried on input of a derivative of the challenge ciphertext in the security proof, the homomorphic operation may result in a ciphertext containing a vector outside the language $\mathcal{L}_\rho$. Since the oracle has to simulate a proof for this vector, each homomorphic evaluation can carry a proof for a potentially false statement. In some sense, each output of the evaluation oracle can be seen as yet another challenge ciphertext. In this setting, our efficient unbounded simulation-sound QA-NIZK proof system comes in handy.

It remains to make sure that CCA1 security is always preserved, should the adversary obtain the evaluation key SK_h at the outset of the game. To this end, we include a second derived one-time homomorphic signature (Z, R, U) in the ciphertext without including its private key in SK_h.

Keygen(λ, t, N)**:** Choose bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$. Then, do the following.

1. Pick $f, g, h \xleftarrow{R} \mathbb{G}$, $x_0, x_1, x_2 \xleftarrow{R} \mathbb{Z}_p$ and set $X_1 = f^{x_1} g^{x_0} \in \mathbb{G}$, $X_2 = h^{x_2} g^{x_0} \in \mathbb{G}$. Then, define $\boldsymbol{f} = (f, 1, g) \in \mathbb{G}^3$ and $\boldsymbol{h} = (1, h, g) \in \mathbb{G}^3$.
2. Choose random polynomials $P_1[Z], P_2[Z], P[Z] \in \mathbb{Z}_p[Z]$ of degree $t-1$ such that $P_1(0) = x_1$, $P_2(0) = x_2$ and $P(0) = x_0$. For each $i \in \{1, \ldots, N\}$, compute $VK_i = (Y_{i,1}, Y_{i,2})$ where $Y_{i,1} = f^{P_1(i)} g^{P(i)}$ and $Y_{i,2} = h^{P_2(i)} g^{P(i)}$.
3. Choose $f_{r,1}, f_{r,2} \xleftarrow{R} \mathbb{G}$ in order to define vectors $\boldsymbol{f}_{r,1} = (f_{r,1}, 1, g)$, $\boldsymbol{f}_{r,2} = (1, f_{r,2}, g)$ and $\boldsymbol{f}_{r,3} = \boldsymbol{f}_{r,1}^{\phi_1} \cdot \boldsymbol{f}_{r,2}^{\phi_2} \cdot (1, 1, g)^{-1}$, where $\phi_1, \phi_2 \xleftarrow{R} \mathbb{Z}_p$. These vectors will be used as a Groth-Sahai CRS for the generation of NIZK proofs showing the validity of decryption shares.
4. Choose a strongly unforgeable one-time signature $\Sigma = (\mathcal{G}, \mathcal{S}, \mathcal{V})$ with verification keys consisting of L-bit strings, for some $L \in \mathsf{poly}(\lambda)$.
5. Generate a key pair for the one-time linearly homomorphic structure-preserving signature of Section 2.5 with $n = 3$. Let $\mathsf{pk}_{ot} = \big(G_z, G_r, H_z, H_u, \{(G_i, H_i)\}_{i=1}^3\big)$ be the public key and let $\mathsf{sk}_{ot} = \{(\varphi_i, \vartheta_i, \varpi_i)\}_{i=1}^3$ be the corresponding private key.
6. Generate one-time homomorphic signatures $\{(Z_j, R_j, U_j)\}_{j=1,2}$ on the vectors $\boldsymbol{f} = (f, 1, g)$ and $\boldsymbol{h} = (1, h, g)$. These signatures consist of $(Z_1, R_1, U_1) = \big(f^{-\varphi_1} g^{-\varphi_3}, f^{-\vartheta_1} g^{-\vartheta_3}, f^{-\varpi_1} g^{-\varpi_3}\big)$ and $(Z_2, R_2, U_2) = \big(h^{-\varphi_2} g^{-\varphi_3}, h^{-\vartheta_2} g^{-\vartheta_3}, h^{-\varpi_2} g^{-\varpi_3}\big)$ and erase sk_{ot}.

7. Generate a key pair $(\mathsf{pk}_{rand}, \mathsf{sk}_{rand})$ as in step 1 of the proof system in Section 3 with $n = 3$. Let $\mathsf{sk}_{rand} = \{(\chi_i, \gamma_i, \delta_i)\}_{i=1}^3$ be the private key for which the corresponding public key is

$$\mathsf{pk}_{rand} = \Big(g_z,\ g_r,\ h_z,\ h_u,\ \{(g_i, h_i)\}_{i=1}^3,\ \mathbf{f} = \big(\boldsymbol{f}_1, \boldsymbol{f}_2, \{\boldsymbol{f}_{3,i}\}_{i=0}^L\big) \Big).$$

8. Use sk_{rand} to generate one-time linearly homomorphic signatures $\{(z_j, r_j, u_j)\}_{j=1,2}$ on the independent vectors $\boldsymbol{f} = (f, 1, g) \in \mathbb{G}^3$ and $\boldsymbol{h} = (1, h, g) \in \mathbb{G}^3$. These are obtained as

$$(z_1, r_1, u_1) = \big(f^{-\chi_1} g^{-\chi_3}, f^{-\gamma_1} g^{-\gamma_3}, f^{-\delta_1} g^{-\delta_3}\big), \quad (z_2, r_2, u_2) = \big(h^{-\chi_2} g^{-\chi_3}, h^{-\gamma_2} g^{-\gamma_3}, h^{-\delta_2} g^{-\delta_3}\big).$$

9. The public key is defined to be

$$PK = \Big(g,\ \boldsymbol{f},\ \boldsymbol{h},\ \boldsymbol{f}_{r,1},\ \boldsymbol{f}_{r,2},\ \boldsymbol{f}_{r,3},\ X_1,\ X_2,\ \mathsf{pk}_{ot},\ \mathsf{pk}_{rand},\ \{(Z_j, R_j, U_j)\}_{j=1}^2\ \{(z_j, r_j, u_j)\}_{j=1}^2\Big).$$

The evaluation key is $SK_h = \mathsf{sk}_{rand} = \{(\chi_i, \gamma_i, \delta_i)\}_{i=1}^3$ while the i-th decryption key share is defined to be $SK_{d,i} = (P_1(i), P_2(i), P(i))$. The vector of verification keys is defined as $\mathbf{VK} = (VK_1, \ldots, VK_N)$, where $VK_i = (Y_{i,1}, Y_{i,2})$ for $i = 1$ to N.

Encrypt(M, PK)**:** to encrypt $M \in \mathbb{G}$, generate a one-time signature key pair $(\mathsf{SVK}, \mathsf{SSK}) \leftarrow \mathcal{G}(\lambda)$.

1. Choose $\theta_1, \theta_2 \xleftarrow{R} \mathbb{Z}_p$ and compute

$$C_0 = M \cdot X_1^{\theta_1} \cdot X_2^{\theta_2}, \qquad C_1 = f^{\theta_1}, \qquad C_2 = h^{\theta_2}, \qquad C_3 = g^{\theta_1 + \theta_2}.$$

2. Construct a first linearly homomorphic signature (Z, R, U) on the vector $(C_1, C_2, C_3) \in \mathbb{G}^3$. Namely, compute $Z = Z_1^{\theta_1} \cdot Z_2^{\theta_2}$, $R = R_1^{\theta_1} \cdot R_2^{\theta_2}$ and $U = U_1^{\theta_1} \cdot U_2^{\theta_2}$.
3. Using $\{(z_j, r_j, u_j)\}_{j=1,2}$, derive another homomorphic signature (z, r, u) on (C_1, C_2, C_3). Namely, compute $z = z_1^{\theta_1} \cdot z_2^{\theta_2}$, $r = r_1^{\theta_1} \cdot r_2^{\theta_2}$ and $u = u_1^{\theta_1} \cdot u_2^{\theta_2}$.
4. Using $\mathsf{SVK} = \mathsf{SVK}[1] \ldots \mathsf{SVK}[L] \in \{0,1\}^L$, define the vector $\boldsymbol{f}_{\mathsf{SVK}} = \boldsymbol{f}_{3,0} \cdot \prod_{i=1}^L \boldsymbol{f}_{3,i}^{\mathsf{SVK}[i]}$ and assemble a Groth-Sahai CRS $\mathbf{f}_{\mathsf{SVK}} = (\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{f}_{\mathsf{SVK}})$. Using $\mathbf{f}_{\mathsf{SVK}}$, generate commitments $\boldsymbol{C}_z$, $\boldsymbol{C}_r$, $\boldsymbol{C}_u$ to the components of $(z, r, u) \in \mathbb{G}^3$ along with proofs $(\boldsymbol{\pi}_1, \boldsymbol{\pi}_2)$ as in step 2 of the proving algorithm of Section 3. Let $(\boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2) \in \mathbb{G}^{15}$ be the resulting NIWI proof.
5. Generate $\sigma = \mathcal{S}(\mathsf{SSK}, (C_0, C_1, C_2, C_3, Z, R, U, \boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2))$ and output

$$C = (\mathsf{SVK}, C_0, C_1, C_2, C_3, Z, R, U, \boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \sigma) \tag{3}$$

Ciphertext-Verify$\big(PK, C\big)$**:** parse C as in (3). Return 1 if and only if these conditions are satisfied: (i) $\mathcal{V}(\mathsf{SVK}, (C_0, C_1, C_2, C_3, Z, R, U, \boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2), \sigma) = 1$; (ii) $(Z, R, U) \in \mathbb{G}^3$ is a valid homomorphic signature on (C_1, C_2, C_3); (iii) $(\boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2) \in \mathbb{G}^{15}$ is a valid proof w.r.t. the CRS $(\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{f}_{\mathsf{SVK}})$ that committed (z, r, u) satisfy the relations (1) for the vector $(C_1, C_2, C_3) \in \mathbb{G}^3$. Here, we define $\boldsymbol{f}_{\mathsf{SVK}} = \boldsymbol{f}_{3,0} \cdot \prod_{i=1}^L \boldsymbol{f}_{3,i}^{\mathsf{SVK}[i]}$.

Share-Decrypt$(PK, i, SK_{d,i}, C)$**:** on inputs $SK_{d,i} = (P_1(i), P_2(i), P(i)) \in \mathbb{Z}_p^3$ and C, return $(i, \perp)$ if **Ciphertext-Verify**$\big(PK, C\big) = 0$. Otherwise, compute $\hat{\mu}_i = \big(\nu_i, \boldsymbol{C}_{P_1}, \boldsymbol{C}_{P_2}, \boldsymbol{C}_P, \pi_{\mu_i}\big)$ which consists of a partial decryption $\nu_i = C_1^{P_1(i)} \cdot C_2^{P_2(i)} \cdot C_3^{P(i)}$ as well as commitments $\boldsymbol{C}_{P_1}, \boldsymbol{C}_{P_2}, \boldsymbol{C}_P$ to exponents $P_1(i), P_2(i), P(i) \in \mathbb{Z}_p$ and a proof π_{ν_i} that these satisfy the equations

$$\nu_i = C_1^{P_1(i)} \cdot C_2^{P_2(i)} \cdot C_3^{P(i)}, \qquad Y_{i,1} = f^{P_1(i)} g^{P(i)}, \qquad Y_{i,2} = h^{P_2(i)} g^{P(i)}.$$

The commitments $\boldsymbol{C}_{P_1}, \boldsymbol{C}_{P_2}, \boldsymbol{C}_P$ and the proof π_{ν_i} are generated using the CRS $(\boldsymbol{f}_{r,1}, \boldsymbol{f}_{r,2}, \boldsymbol{f}_{r,3})$ (see Appendix A for details). Then, return $\mu_i = (i, \hat{\mu}_i)$.

Share-Verify$\big(PK, VK_i, C, (i, \hat{\mu}_i)\big)$**:** parse C as in (3) and VK_i as $(Y_{i,1}, Y_{i,2})$. If $\hat{\mu}_i = \perp$ or $\hat{\mu}_i$ cannot be parsed as $\big(\nu_i, \boldsymbol{C}_{P_1}, \boldsymbol{C}_{P_2}, \boldsymbol{C}_P, \pi_{\mu_i}\big)$, return 0. Otherwise, return 1 if and only if π_{μ_i} is valid.

Combine$(PK, \mathbf{VK}, C, \{(i, \hat{\mu}_i)\}_{i\in S})$**:** for each $i \in S$, parse the share $\hat{\mu}_i$ as $\big(\nu_i, \boldsymbol{C}_{P_1}, \boldsymbol{C}_{P_2}, \boldsymbol{C}_P, \pi_{\mu_i}\big)$ and return $\perp$ if **Share-Verify**$\big(PK, C, (i, \hat{\mu}_i)\big) = 0$. Otherwise, compute $\nu = \prod_{i\in S} \nu_i^{\Delta_{i,S}(0)}$, which equals $\nu = C_1^{x_1} \cdot C_2^{x_2} \cdot C_3^{x_0} = X_1^{\theta_1} \cdot X_2^{\theta_2}$ and in turn reveals $M = C_0/\nu$.

Eval$(PK, SK_h, C^{(1)}, C^{(2)})$**:** parse SK_h as $\{(\chi_i, \gamma_i, \delta_i)\}_{i=1}^3$. For each $j \in \{1, 2\}$, parse $C^{(j)}$ as

$$C^{(j)} = (\mathsf{SVK}^{(j)}, C_0^{(j)}, C_1^{(j)}, C_2^{(j)}, C_3^{(j)}, Z^{(j)}, R^{(j)}, U^{(j)}, \boldsymbol{C}_z^{(j)}, \boldsymbol{C}_r^{(j)}, \boldsymbol{C}_u^{(j)}, \boldsymbol{\pi}_1^{(j)}, \boldsymbol{\pi}_2^{(j)}, \sigma^{(j)})$$

and return $\perp$ if either $C^{(1)}$ or $C^{(2)}$ is invalid. Otherwise,

1. Compute $C_0 = \prod_{j=1}^2 C_0^{(j)}$, $C_1 = \prod_{j=1}^2 C_1^{(j)}$, $C_2 = \prod_{j=1}^2 C_2^{(j)}$ and $C_3 = \prod_{j=1}^2 C_3^{(j)}$ as well as $Z = \prod_{j=1}^2 Z^{(j)}$, $R = \prod_{j=1}^2 R^{(j)}$ and $U = \prod_{j=1}^2 U^{(j)}$.
2. Generate a new one-time signature key pair $(\mathsf{SVK}, \mathsf{SSK}) \leftarrow \mathcal{G}(\lambda)$. Using $SK_h = \{(\chi_i, \gamma_i, \delta_i)\}_{i=1}^3$, generate proof elements $\boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2$ on the vector (C_1, C_2, C_3) using the simulator of the proof system in Section 3 with the one-time verification key SVK.
3. Return the derived ciphertext $C = (\mathsf{SVK}, C_0, C_1, C_2, C_3, Z, R, U, \boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \sigma)$ where $\sigma = \mathcal{S}(\mathsf{SSK}, (C_0, C_1, C_2, C_3, Z, R, U, \boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2))$.

In Appendix H, we prove the KH-CCA security of the scheme assuming that Σ is a strongly unforgeable one-time signature and that the DLIN assumption holds in $\mathbb{G}$.

In some applications, it may be desirable to add an extra randomization step to the evaluation algorithm in order to make sure that derived ciphertexts will be indistinguishable from freshly generated encryption (similarly to [48]). It is straightforward to modify the scheme to obtain this property.

If the scheme is instantiated using Groth's one-time signature [32], the ciphertext consists of 25 elements of $\mathbb{G}$ and two elements of $\mathbb{Z}_p$. It is interesting to compare the above system with an instantiation of the same design principle using the best known Groth-Sahai-based unbounded simulation-sound proof [15][Appendix A.2], which requires 65 group elements in this specific case. With this proof system, we end up with 77 group elements per ciphertexts under the DLIN assumption (assuming that an element of $\mathbb{Z}_p$ has the same length as the representation of a group element). The above realization thus saves 50 group elements and compresses ciphertexts to 35% of their original length.

We note that it is possible to adapt the scheme to rely on the Symmetric eXternal Diffie-Hellman assumption in asymmetric pairings $e : \mathbb{G} \times \hat{\mathbb{G}} \to \mathbb{G}_T$. In this case, the ciphertext contains 9 elements of $\mathbb{G}$, 2 elements of $\hat{\mathbb{G}}$ and a one-time key pair (SVK, σ). Using the one-time signature of [32], the ciphertext overhead amounts to 4096 bits on Barreto-Naehrig curves [4] if each element of $\mathbb{G}$ (resp. each element of $\hat{\mathbb{G}}$) has a 256-bit (resp. 512-bit) representation.

References

1. M. Abe, S. Fehr. Adaptively Secure Feldman VSS and Applications to Universally-Composable Threshold Cryptography. In *Crypto'04*, *LNCS* 3152, pp. 317–334, 2004.
2. M. Abe, G. Fuchsbauer, J. Groth, K. Haralambiev, M. Ohkubo. Structure-Preserving Signatures and Commitments to Group Elements. In *Crypto'10*, *LNCS* 6223, pp. 209–236, 2010.
3. M.Abe, K. Haralambiev, M. Ohkubo. Signing on Elements in Bilinear Groups for Modular Protocol Design. In Cryptology ePrint Archive: Report 2010/133, 2010.
4. P. Barreto, M. Naehrig. Pairing-Friendly Elliptic Curves of Prime Order. In *SAC'05*, *LNCS* 3897, pp. 319–331, 2005.
5. M. Belenkiy, J. Camenisch, M. Chase, M. Kohlweiss, A. Lysyanskaya, H. Shacham. Randomizable Proofs and Delegatable Anonymous Credentials. In *Crypto'09*, *LNCS* 5677, pp. 108–125, 2009.
6. M. Bellare, T. Ristenpart. Simulation without the Artificial Abort: Simplified Proof and Improved Concrete Security for Waters' IBE Scheme. In *Eurocrypt'09*, *LNCS* 5479, pp. 407–424, 2009.
7. M. Bellare, P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM CCS*, pp. 62–73, 1993.
8. M. Blum, P. Feldman, S. Micali. Non-Interactive Zero-Knowledge and Its Applications. In *STOC'88*, pp. 103-112, 1988.

9. D. Boneh, X. Boyen, S. Halevi. Chosen Ciphertext Secure Public Key Threshold Encryption Without Random Oracles. In *CT-RSA'06*, *LNCS* 3860, pp. 226–243, 2006.
10. D. Boneh, X. Boyen, H. Shacham. Short group signatures. In *Crypto'04*, LNCS 3152, pp. 41–55, 2004.
11. D. Boneh, M. Hamburg, S. Halevi, R. Ostrovsky. Circular-Secure Encryption from Decision Diffie-Hellman. In *Crypto'08*, LNCS 5157, pp. 108–125, 2008.
12. D. Boneh, G. Segev, B. Waters. Targeted malleability: homomorphic encryption for restricted computations. In *ITCS 2012*, pp. 350–366, 2012.
13. C. Boyd. Digital Multisignatures. In *Cryptography and Coding*, Oxford University Press, pp. 241–246, 1989.
14. X. Boyen, Q. Mei, B. Waters. Direct Chosen Ciphertext Security from Identity-Based Techniques. in *ACM CCS'05*, pp. 320–329, 2005.
15. J. Camenisch, N. Chandran, and V. Shoup. A public key encryption scheme secure against key dependent chosen plaintext and adaptive chosen ciphertext attacks. In *Eurocrypt'09*, LNCS 5479, pp. 351–368, 2009.
16. R. Canetti, R. Gennaro, S. Jarecki, H. Krawczyk, T. Rabin. Adaptive Security for Threshold Cryptosystems. In *Crypto'99*, *LNCS* 1666, pp. 98–115, 1999.
17. J. Cathalo, B. Libert, M. Yung. Group Encryption: Non-Interactive Realization in the Standard Model. In *Asiacrypt'09*, *LNCS* 5912, pp. 179–196, 2009.
18. M. Chase, M. Kohlweiss, A. Lysyanskaya, S. Meiklejohn. Malleable Proof Systems and Applications. In *Eurocrypt'12*, *LNCS* 7237, pp. 281–300, 2012.
19. J.-S. Coron, T. Lepoint, M. Tibouchi. Practical Multilinear Maps over the Integers. In *Crypto'13*, *LNCS* 8042, pp. 476–493, 2013.
20. R. Cramer, V. Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *Crypto'98*, LNCS 1462, pp. 13–25, 1998.
21. R. Cramer, V. Shoup. Universal Hash Proofs and a Paradigm for Adaptive Chosen Ciphertext Secure Public-Key Encryption. In *Eurocrypt'02*, *LNCS* 2332, pp. 45–64, 2002.
22. Y. Desmedt. Society and Group Oriented Cryptography: A New Concept. In *Crypto'87*, *LNCS* 293, pp. 120–127, 1987.
23. Y. Desmedt, Y. Frankel. Threshold Cryptosystems. In *Crypto'89*, *LNCS* 435, pp. 307–315, 1989.
24. K. Emura, G. Hanaoka, G. Ohtake, T. Matsuda, S. Yamada. Chosen Ciphertext Secure Keyed-Homomorphic Public-Key Encryption. In *PKC'13*, *LNCS* 7778, pp. 32–50, 2013.
25. A. Escala, G. Herold, E. Kiltz, C. Ràfols, J. Villar. An Algebraic Framework for Diffie-Hellman Assumptions. In *Crypto'13*, *LNCS* 8043, pp. 129–147, 2013.
26. A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Crypto'86*, LNCS 263, pages 186–194, 1986.
27. P.-A. Fouque, D. Pointcheval. Threshold Cryptosystems Secure against Chosen-Ciphertext Attacks. In *Asiacrypt'01*, *LNCS* 2248, pp. 351–368, 2001.
28. Y. Frankel, P. MacKenzie, M. Yung. Adaptively-Secure Distributed Public-Key Systems. In *ESA'99*, *LNCS* 1643, pp. 4–27, 1999.
29. S. Garg, C. Gentry, S. Halevi. Candidate Multilinear Maps from Ideal Lattices. In *Eurocrypt'13*, *LNCS* 7881, pp. 1–17, 2013.
30. C. Gentry, D. Wichs. Separating succinct non-interactive arguments from all falsifiable assumptions. In *STOC'11*, pp. 99-108, 2011.
31. J. Groth, R. Ostrovsky, and A. Sahai. Perfect non-interactive zero knowledge for NP. In *Eurocrypt'06*, *LNCS* 4004, pp. 339–358, 2006.
32. J. Groth. Simulation-sound NIZK proofs for a practical language and constant size group signatures. In *Asiacrypt'06*, *LNCS* 4284, pp. 444–459, 2006.
33. J. Groth, A. Sahai. Efficient non-interactive proof systems for bilinear groups. In *Eurocrypt'08*, LNCS 4965, pp. 415–432, 2008.
34. D. Hofheinz, T. Jager. Tightly Secure Signatures and Public-Key Encryption. In *Crypto'12*, *LNCS* 7417, pp. 590–607, 2012.
35. D. Hofheinz, E. Kiltz. Programmable Hash Functions and Their Applications. In *Crypto'08*, *LNCS* 5157, pp. 21–38, 2008.
36. S. Jarecki, A. Lysyanskaya. Adaptively Secure Threshold Cryptography: Introducing Concurrency, Removing Erasures. In *Eurocrypt'00*, *LNCS* 1807, pp. 221–242, 2000.
37. C. Jutla, A. Roy. Relatively-Sound NIZKs and Password-Based Key-Exchange. In *PKC'12*, *LNCS* 7293, pp. 485–503, 2012.
38. C. Jutla, A. Roy. Shorter Quasi-Adaptive NIZK Proofs for Linear Subspaces. In *Asiacrypt'13*, *LNCS* series, 2013. Cryptology ePrint Archive: Report 2013/109, 2013.
39. J. Katz, V. Vaikuntanathan. Round-Optimal Password-Based Authenticated Key Exchange. In *TCC'11*, LNCS 6597, pp. 293–310, 2011.
40. A. Lewko. Tools for Simulating Features of Composite Order Bilinear Groups in the Prime Order Setting. In *Eurocrypt'12*, *LNCS* 7237, pp. 318–335, 2012.
41. B. Libert, M. Yung. Adaptively Secure Non-Interactive Threshold Cryptosystems. In *ICALP 2011*, LNCS 6756, pp. 588–600, 2011.

42. B. Libert, M. Yung. Non-Interactive CCA2-Secure Threshold Cryptosystems with Adaptive Security: New Framework and Constructions. In *TCC 2012*, LNCS 7194, pp. 75–93, 2012.
43. B. Libert, T. Peters, M. Joye, M. Yung. Linearly Homomorphic Structure-Preserving Signatures and their Applications. In *Crypto 2013*, *LNCS* 8043, pp. 289–307, 2013.
44. T. Malkin, I. Teranishi, Y. Vahlis, M. Yung. Signatures resilient to continual leakage on memory and computation. In *TCC'11*, LNCS 6597, pp. 89–106, 2011.
45. M. Naor. On cryptographic assumptions and challenges. In *Crypto'03*, *LNCS* 2729, pp. 96–109, 2003.
46. M. Naor, M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *STOC'90*, pp. 427–437, 1990.
47. M. Prabhakaran, M. Rosulek. Rerandomizable RCCA Encryption. *Crypto 2007*, *LNCS* 4622, pp. 517–534, 2007.
48. M. Prabhakaran, M. Rosulek. Homomorphic Encryption with CCA Security. *ICALP 2008*, *LNCS* 5126, pp. 667–678, 2008.
49. M. Prabhakaran, M. Rosulek. Towards Robust Computation on Encrypted Data. *Asiacrypt 2008*, *LNCS* 5350, pp. 216–233, 2008.
50. C. Rackoff, D. Simon. Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack. In *Crypto'91*, *LNCS* 576, pp. 433–444, 1991.
51. A. Sahai. Non-Malleable Non-Interactive Zero Knowledge and Adaptive Chosen-Ciphertext Security. In *FOCS'99*, pp. 543–553, 1999.
52. H. Shacham. A Cramer-Shoup encryption scheme from the linear assumption and from progressively weaker linear variants. Cryptology ePrint Archive: Report 2007/074, 2007.
53. V. Shoup, R. Gennaro. Securing Threshold Cryptosystems against Chosen Ciphertext Attack. In *J. of Cryptology*, 15(2), pp. 75–96, 2002. Earlier version in *Eurocrypt'98*, *LNCS* 1403, pp. 1–16, 1998.
54. B. Waters. Efficient Identity-Based Encryption Without Random Oracles. In *Eurocrypt'05*, *LNCS* 3494, 2005.
55. H. Wee. Threshold and Revocation Cryptosystems via Extractable Hash Proofs. In *Eurocrypt'11*, *LNCS* 6632, pp. 589–609, 2011.

A Groth-Sahai Proofs

In the notations of this section, when vectors $\boldsymbol{A}$ and $\boldsymbol{B}$ are vectors of group elements, $\boldsymbol{A} \cdot \boldsymbol{B}$ denotes their entry-wise product.

Under the DLIN assumption in symmetric bilinear groups $(\mathbb{G}, \mathbb{G}_T)$, the Groth-Sahai (GS) proof systems [33] use a CRS consisting of three vectors $\boldsymbol{g}_1, \boldsymbol{g}_2, \boldsymbol{g}_3 \in \mathbb{G}^3$, where $\boldsymbol{g}_1 = (g_1, 1, g)$, $\boldsymbol{g}_2 = (1, g_2, g)$ for some $g_1, g_2 \in_R \mathbb{G}$. In this setting, a commitment to a group element $X \in \mathbb{G}$ is computed as $\boldsymbol{C} = (1, 1, X) \cdot \boldsymbol{g_1}^r \cdot \boldsymbol{g_2}^s \cdot \boldsymbol{g_3}^t$ with $r, s, t \xleftarrow{R} \mathbb{Z}_p^*$. In order to obtain perfectly sound proofs, $\boldsymbol{g}_3$ is chosen as $\boldsymbol{g_3} = \boldsymbol{g_1}^{\xi_1} \cdot \boldsymbol{g_2}^{\xi_2}$, with $\xi_1, \xi_2 \xleftarrow{R} \mathbb{Z}_p^*$, so that the commitment $\boldsymbol{C} = (g_1^{r+\xi_1 t}, g_2^{s+\xi_2 t}, X \cdot g^{r+s+t(\xi_1+\xi_2)})$ is extractable as it is distributed as a Boneh-Boyen-Shacham (BBS) ciphertext [10] that can be decrypted using the discrete logarithms $\alpha_1 = \log_g(g_1)$, $\alpha_2 = \log_g(g_2)$. In order to switch to the witness indistinguishability setting, the vectors $\boldsymbol{g}_1, \boldsymbol{g}_2, \boldsymbol{g}_3$ must be linearly independent so as to span the entire space where $\boldsymbol{C}$ lives and make sure that $\boldsymbol{C}$ is a perfectly hiding commitment. Under the DLIN assumption, the two kinds of reference strings are computationally indistinguishable.

When it comes to commit to an exponent $x \in \mathbb{Z}_p$, the prover computes $\boldsymbol{C} = \boldsymbol{\varphi}^x \cdot \boldsymbol{g}_1^r \cdot \boldsymbol{g}_2^s$, with $r, s \xleftarrow{R} \mathbb{Z}_p^*$, using a CRS comprising vectors $\boldsymbol{\varphi}, \boldsymbol{g}_1, \boldsymbol{g}_2$. In the soundness setting $\boldsymbol{\varphi}, \boldsymbol{g}_1, \boldsymbol{g}_2$ are linearly independent vectors while, in the perfect WI setting, choosing $\boldsymbol{\varphi}$ in $\mathrm{span}(\boldsymbol{g}_1, \boldsymbol{g}_2)$ yields a perfectly hiding commitment as $\boldsymbol{C}$ is always a BBS encryption of $1_{\mathbb{G}}$.

To prove that committed group elements or exponents satisfy certain relations, the Groth-Sahai methodology [33] requires one commitment per variable and one proof element per relation. Efficient NIWI proofs are available for multi-exponentiation equations of the form

$$\prod_{i=1}^{m} \mathcal{A}_i^{y_i} \cdot \prod_{j=1}^{n} \mathcal{X}_j^{b_j} \cdot \prod_{i=1}^{m} \cdot \prod_{j=1}^{n} \mathcal{X}_j^{y_i \gamma_{ij}} = T,$$

for variables $\mathcal{X}_1, \ldots, \mathcal{X}_n \in \mathbb{G}$, $y_1, \ldots, y_m \in \mathbb{Z}_p$ and constants $T, \mathcal{A}_1, \ldots, \mathcal{A}_m \in \mathbb{G}$, $b_1, \ldots, b_n \in \mathbb{Z}_p$ and $\gamma_{ij} \in \mathbb{Z}_p$, for $i \in \{1, \ldots, m\}, j \in \{1, \ldots, n\}$.

For linear equations (*i.e.*, where $\gamma_{ij} = 0$ for all i, j) depends on the form of the considered equation.

Namely, linear multi-exponentiation equations of the type $\prod_{j=1}^{n} \mathcal{X}_j^{b_j} = T$ (resp. $\prod_{i=1}^{m} \mathcal{A}_i^{y_i} = T$) demand 3 (resp. 2) group elements.

Multi-exponentiation equations admit NIZK proofs. On a simulated CRS, the representation $(\xi_1, \xi_2) \in \mathbb{Z}_p^2$ of $\boldsymbol{\varphi}$ as $\boldsymbol{\varphi} = \boldsymbol{g}_1^{\xi_1} \cdot \boldsymbol{g}_2^{\xi_2}$ can serve as a trapdoor that makes it possible to perfectly simulate proofs without knowing the witnesses.

Efficient NIWI proofs also exist for pairing-product relations, which are the form

$$\prod_{i=1}^{n} e(\mathcal{A}_i, \mathcal{X}_i) \cdot \prod_{i=1}^{n} \cdot \prod_{j=1}^{n} e(\mathcal{X}_i, \mathcal{X}_j)^{a_{ij}} = t_T,$$

for variables $\mathcal{X}_1, \ldots, \mathcal{X}_n \in \mathbb{G}$ and constants $t_T \in \mathbb{G}_T$, $\mathcal{A}_1, \ldots, \mathcal{A}_n \in \mathbb{G}$, $a_{ij} \in \mathbb{Z}_p$, for $i, j \in \{1, \ldots, n\}$. For each linear pairing product equation (where $a_{ij} = 0$ for all i, j), a proof fits within 3 group elements. Quadratic equations are somewhat more space-consuming and take 9 group elements each. At the cost of introducing extra variables, pairing product equations can also have NIZK proofs.

B Definitions for Linearly Homomorphic Structure-Preserving Signatures

Let $(\mathbb{G}, \mathbb{G}_T)$ be groups of prime order p such that a bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ can be efficiently computed.

A signature scheme is *structure-preserving* [3, 2] if messages, signatures and public keys all live in the group $\mathbb{G}$. In linearly homomorphic structure-preserving signatures, the message space $\mathcal{M}$ consists of pairs $\mathcal{M} := \mathcal{T} \times \mathbb{G}^n$, for some $n \in \mathbb{N}$, where $\mathcal{T}$ is a tag space. Depending on the application, one may want the tags to be group elements or not. In this paper, they can be arbitrary strings.

Definition 4. *A linearly homomorphic structure-preserving signature scheme over $(\mathbb{G}, \mathbb{G}_T)$ is a tuple of efficient algorithms $\Sigma = (\mathsf{Keygen}, \mathsf{Sign}, \mathsf{SignDerive}, \mathsf{Verify})$ for which the message space consists of $\mathcal{M} := \mathcal{T} \times \mathbb{G}^n$, for some integer $n \in \mathsf{poly}(\lambda)$ and some set $\mathcal{T}$, and with the following specifications.*

$\mathsf{Keygen}(\lambda, n)$: *is a randomized algorithm that takes in a security parameter $\lambda \in \mathbb{N}$ and an integer $n \in \mathsf{poly}(\lambda)$ denoting the dimension of vectors to be signed. It outputs a key pair $(\mathsf{pk}, \mathsf{sk})$, where pk includes the description of a tag space $\mathcal{T}$, where each tag serves as a file identifier.*

$\mathsf{Sign}(\mathsf{sk}, \tau, \boldsymbol{M})$: *is a possibly randomized algorithm that takes as input a private key sk, a file identifier $\tau \in \mathcal{T}$ and a vector $\boldsymbol{M} = (M_1, \ldots, M_n) \in \mathbb{G}^n$. It outputs a signature $\sigma \in \mathbb{G}^{n_s}$, for some $n_s \in \mathsf{poly}(\lambda)$.*

$\mathsf{SignDerive}(\mathsf{pk}, \tau, \{(\omega_i, \sigma^{(i)})\}_{i=1}^{\ell})$: *is a (possibly randomized) derivation algorithm. It inputs a public key pk, a file identifier τ as well as ℓ pairs $(\omega_i, \sigma^{(i)})$, each of which consists of a coefficient $\omega_i \in \mathbb{Z}_p$ and a signature $\sigma^{(i)} \in \mathbb{G}^{n_s}$. It outputs a signature $\sigma \in \mathbb{G}^{n_s}$ on the vector $\boldsymbol{M} = \prod_{i=1}^{\ell} \boldsymbol{M}_i{}^{\omega_i}$, where $\sigma^{(i)}$ is a signature on $\boldsymbol{M}_i$.*

$\mathsf{Verify}(\mathsf{pk}, \tau, \boldsymbol{M}, \sigma)$: *is a deterministic verification algorithm that takes as input a public key pk, a file identifier $\tau \in \mathcal{T}$, a signature σ and a vector $\boldsymbol{M} = (M_1, \ldots, M_n)$. It outputs 0 or 1 depending on whether σ is deemed valid or not.*

In a *one-time* linearly homomorphic SPS, the tag τ can be omitted in the specification as a given key pair $(\mathsf{pk}, \mathsf{sk})$ only allows signing one linear subspace.

As in all linearly homomorphic signatures, the security requirement is that the adversary be unable to create a valid triple $(\tau^\star, \boldsymbol{M}^\star, \sigma^\star)$ for a new file identifier $\tau^\star$ or, if $\tau^\star$ is recycled from one or more honestly generated signatures, for a vector $\boldsymbol{M}^\star$ outside the linear span of the vectors that have been legitimately signed for the tag $\tau^\star$.

An important property is that the $\mathsf{SignDerive}$ algorithm must operate on vectors that are all labeled with the same tag.

C Randomizable Linearly Homomorphic Structure-Preserving Signatures

This section recalls the randomizable linearly homomorphic structure-preserving signature of [43].

In the scheme, each signature basically consists of a Groth-Sahai NIWI proof of knowledge of a one-time signature (z, r, u) on the signed vector $(M_1, \ldots, M_n)$. This proof of knowledge is generated for a Groth-Sahai CRS which depends on the tag that identifies the subspace being signed.

In the following notations, for each $h \in \mathbb{G}$ and any vector $\boldsymbol{g} = (g_1, g_2, g_3) \in \mathbb{G}^3$, we denote by $E(h, \boldsymbol{g})$ the vector $(e(h, g_1), e(h, g_2), e(h, g_3)) \in \mathbb{G}_T^3$.

Keygen(λ, n): given a security parameter λ and the dimension $n \in \mathbb{N}$ of the subspace to be signed, choose bilinear group $(\mathbb{G}, \mathbb{G}_T)$ of order $p > 2^\lambda$ with a generator $g \xleftarrow{R} \mathbb{G}$ as well as $g_z, g_r, h_z, h_u \xleftarrow{R} \mathbb{G}$ and do the following.

1. For $i = 1$ to n, pick $\chi_i, \gamma_i, \delta_i \xleftarrow{R} \mathbb{Z}_p$ and compute $g_i = {g_z}^{\chi_i} {g_r}^{\gamma_i}$ and $h_i = {h_z}^{\chi_i} {h_u}^{\delta_i}$.
2. Generate $L+1$ Groth-Sahai common reference strings, where $L \in \mathsf{poly}(\lambda)$ is the length of each tag $\tau \in \mathcal{T} = \{0,1\}^L$. To this end, choose $f_1, f_2 \xleftarrow{R} \mathbb{G}$ and define vectors $\boldsymbol{f}_1 = (f_1, 1, g) \in \mathbb{G}^3$, $\boldsymbol{f}_2 = (1, f_2, g) \in \mathbb{G}^3$. Then, pick $\boldsymbol{f}_{3,i} \xleftarrow{R} \mathbb{G}^3$ for $i = 0$ to L.

The public key consists of

$$\mathsf{pk} = \Big(\ (\mathbb{G}, \mathbb{G}_T),\ g_z,\ g_r,\ h_z,\ h_u,\ \{g_i, h_i\}_{i=1}^n,\ \mathbf{f} = \big(\boldsymbol{f}_1, \boldsymbol{f}_2, \{\boldsymbol{f}_{3,i}\}_{i=0}^L\big) \ \Big)$$

while the private key is $\mathsf{sk} = \big({h_z}^{\alpha_r}, \{\chi_i, \gamma_i, \delta_i\}_{i=1}^n\big)$.

Sign($\mathsf{sk}, \tau, (M_1, \ldots, M_n)$): to sign a vector $(M_1, \ldots, M_n) \in \mathbb{G}^n$ using $\mathsf{sk} = \big({h_z}^{\alpha_r},\ \{\chi_i, \gamma_i, \delta_i\}_{i=1}^n\big)$ with the file identifier τ, conduct the following steps.

1. Choose $\theta \xleftarrow{R} \mathbb{Z}_p$ at random and compute $z = {g_r}^{\theta} \cdot \prod_{i=1}^n M_i^{-\chi_i}$, $r = {g_z}^{-\theta} \cdot \prod_{i=1}^n M_i^{-\gamma_i}$ and $u = {h_z}^{-\theta \cdot \alpha_r} \cdot \prod_{i=1}^n M_i^{-\delta_i}$.
2. Using the bits $\tau[1] \ldots \tau[L]$ of $\tau \in \{0,1\}^L$, define the vector $\boldsymbol{f}_\tau = \boldsymbol{f}_{3,0} \cdot \prod_{i=1}^L \boldsymbol{f}_{3,i}^{\tau[i]}$ so as to assemble a Groth-Sahai CRS $\mathbf{f}_\tau = (\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{f}_\tau)$.
3. Using $\mathbf{f}_\tau = (\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{f}_\tau)$, compute commitments $\boldsymbol{C}_z = (1_\mathbb{G}, 1_\mathbb{G}, z) \cdot \boldsymbol{f}_1^{\nu_{z,1}} \cdot \boldsymbol{f}_2^{\nu_{z,2}} \cdot \boldsymbol{f}_\tau^{\nu_{z,3}}$ and

$$\boldsymbol{C}_r = (1_\mathbb{G}, 1_\mathbb{G}, r) \cdot \boldsymbol{f}_1^{\nu_{r,1}} \cdot \boldsymbol{f}_2^{\nu_{r,2}} \cdot \boldsymbol{f}_\tau^{\nu_{r,3}} \qquad \boldsymbol{C}_u = (1_\mathbb{G}, 1_\mathbb{G}, u) \cdot \boldsymbol{f}_1^{\nu_{u,1}} \cdot \boldsymbol{f}_2^{\nu_{u,2}} \cdot \boldsymbol{f}_\tau^{\nu_{u,3}}$$

to the derived z, r and u, respectively. Generate NIWI proofs $\boldsymbol{\pi}_1 = (\pi_{1,1}, \pi_{1,2}, \pi_{1,3}) \in \mathbb{G}^3$ and $\boldsymbol{\pi}_2 = (\pi_{2,1}, \pi_{2,2}, \pi_{2,3}) \in \mathbb{G}^3$ that (z, r, u) satisfy the verification equations

$$1_{\mathbb{G}_T} = e(g_z, z) \cdot e(g_r, r) \cdot \prod_{i=1}^n e(g_i, M_i) \qquad 1_{\mathbb{G}_T} = e(h_z, z) \cdot e(h_u, u) \cdot \prod_{i=1}^n e(h_i, M_i).$$

These proofs are obtained as

$$\boldsymbol{\pi}_1 = (\pi_{1,1}, \pi_{1,2}, \pi_{1,3}) = \big({g_z}^{-\nu_{z,1}} \cdot {g_r}^{-\nu_{r,1}},\ {g_z}^{-\nu_{z,2}} \cdot {g_r}^{-\nu_{r,2}},\ {g_z}^{-\nu_{z,3}} \cdot {g_r}^{-\nu_{r,3}}\big)$$
$$\boldsymbol{\pi}_2 = (\pi_{2,1}, \pi_{2,2}, \pi_{2,3}) = \big({h_z}^{-\nu_{z,1}} \cdot {h_u}^{-\nu_{u,1}},\ {h_z}^{-\nu_{z,2}} \cdot {h_u}^{-\nu_{u,2}},\ {h_z}^{-\nu_{z,3}} \cdot {h_u}^{-\nu_{u,3}}\big).$$

The signature consists of $\sigma = (\boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2) \in \mathbb{G}^{15}$.

SignDerive($\mathsf{pk}, \tau, \{(\omega_i, \sigma^{(i)})\}_{i=1}^\ell$): given pk, a file identifier τ and ℓ tuples $(\omega_i, \sigma^{(i)})$, parse each signature $\sigma^{(i)}$ as a tuple of the form $\sigma^{(i)} = (\boldsymbol{C}_{z,i}, \boldsymbol{C}_{r,i}, \boldsymbol{C}_{u,i}, \boldsymbol{\pi}_{1,i}, \boldsymbol{\pi}_{2,i}) \in \mathbb{G}^{15}$ for $i = 1$ to ℓ.

1. Compute $\boldsymbol{C}_z = \prod_{i=1}^\ell \boldsymbol{C}_{z,i}^{\omega_i}$, $\boldsymbol{C}_r = \prod_{i=1}^\ell \boldsymbol{C}_{r,i}^{\omega_i}$, $\boldsymbol{C}_u = \prod_{i=1}^\ell \boldsymbol{C}_{u,i}^{\omega_i}$, $\boldsymbol{\pi}_1 = \prod_{i=1}^\ell \boldsymbol{\pi}_{1,i}^{\omega_i}$ as well as $\boldsymbol{\pi}_2 = \prod_{i=1}^\ell \boldsymbol{\pi}_{2,i}^{\omega_i}$.
2. Re-randomize the above commitments and proofs and return $\sigma = (\boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2)$.

Verify($\mathsf{pk}, \boldsymbol{\sigma}, \boldsymbol{\tau}, (\boldsymbol{M_1}, \ldots, \boldsymbol{M_n})$): given $(\tau, (M_1, \ldots, M_n))$ and a purported signature σ, parse σ as $(\boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2)$. Return 1 iff $(M_1, \ldots, M_n) \neq (1_{\mathbb{G}}, \ldots, 1_{\mathbb{G}})$ and the proofs $(\boldsymbol{\pi}_1, \boldsymbol{\pi}_2)$ satisfy

$$\prod_{i=1}^{n} E\big(g_i, (1_{\mathbb{G}}, 1_{\mathbb{G}}, M_i)\big)^{-1} = E\big(g_z, \boldsymbol{C}_z\big) \cdot E\big(g_r, \boldsymbol{C}_r\big) \cdot E(\pi_{1,1}, \boldsymbol{f}_1) \cdot E(\pi_{1,2}, \boldsymbol{f}_2) \cdot E(\pi_{1,3}, \boldsymbol{f}_\tau) \tag{4}$$
$$\prod_{i=1}^{n} E\big(h_i, (1_{\mathbb{G}}, 1_{\mathbb{G}}, M_i)\big)^{-1} = E\big(h_z, \boldsymbol{C}_z\big) \cdot E\big(h_u, \boldsymbol{C}_u\big) \cdot E(\pi_{2,1}, \boldsymbol{f}_1) \cdot E(\pi_{2,2}, \boldsymbol{f}_2) \cdot E(\pi_{2,3}, \boldsymbol{f}_\tau).$$

We remark that the scheme can be simplified by setting $\theta = 0$ in all algorithms: since all non-interactive proofs are generated for a perfectly NIWI Groth-Sahai CRS, this modification does not affect the distribution of signatures whatsoever. In Sections 3 and 5, we use this simplified version of the scheme.

The scheme is only known [43] to be secure in a relaxed model where the adversary is only deemed successful if it additionally provides evidence that its output vector is indeed independent of those for which it obtained signatures with respect to the target tag $\tau^\star$. In our applications, this restriction will not be a problem at all since, in all security proofs, the reduction will always be able to detect when the adversary has won without requiring explicit evidence for it.

D Extensions Based on the k-Linear Assumption

To instantiate our proof systems using the k-linear assumption with $k > 2$, we first need to extend the one-time linearly homomorphic structure-preserving signature of [43]. To this end, we need to define the following assumption which is implied by the k-linear assumption in the same way as SDP is implied by DLIN.

Definition 5. *The* **Simultaneous k-wise Pairing** *(k-SDP) problem is, given a random tuple*

$$(g_{1,z}, \ldots, g_{k,z}, g_{1,r}, \ldots, g_{k,r}) \in_R \mathbb{G}^{2k},$$

to find a non-trivial vector $(z, r_1, \ldots, r_k) \in \mathbb{G}^{k+1}$ such that

$$e(g_{j,z}, z) \cdot e(g_{j,r}, r_j) = 1_{\mathbb{G}_T} \qquad\qquad j \in \{1, \ldots, k\} \tag{5}$$

and $z \neq 1_{\mathbb{G}}$.

Given a k-linear instance $(g_{1,r}, \ldots, g_{k,r}, g_{1,r}^{a_1}, \ldots, g_{k,r}^{a_k}, \eta) \in \mathbb{G}^{2k+1}$, for any non-trivial tuple $(z, r_1, \ldots, r_k)$ satisfying $e(g_{j,r}^{a_j}, z) \cdot e(g_{j,r}, r_j) = 1_{\mathbb{G}_T}$ for each $j \in \{1, \ldots, k\}$, we have

$$\eta = g^{\sum_{j=1}^{k} a_j} \qquad \Leftrightarrow \qquad e(g, \prod_{j=1}^{k} r_j) \cdot e(z, \eta) = 1_{\mathbb{G}_T}.$$

Hence, any algorithm solving k-SDP with non-negligible probability implies a k-linear distinguisher.

Under the k-SDP assumption, the one-time linearly homomorphic structure-preserving signature of [43] can be extended as follows.

Keygen(λ, n): given a security parameter λ and the dimension $n \in \mathbb{N}$ of vectors to be signed, choose bilinear group $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$. For $j = 1$ to k, choose generators $g_{j,z}, g_{j,r} \xleftarrow{R} \mathbb{G}$. Then, for each $i = 1$ to n, $j = 1$ to k, pick $\chi_i \xleftarrow{R} \mathbb{Z}_p$, $\gamma_{j,i} \xleftarrow{R} \mathbb{Z}_p$ and compute $g_{j,i} = g_{j,z}{}^{\chi_i} g_{j,r}{}^{\gamma_{j,i}}$. The private key is $\mathsf{sk} = \big(\{\chi_i, \{\gamma_{j,i}\}_{j=1}^{k}\}_{i=1}^{n}\big)$ while the public key is

$$\mathsf{pk} = \Big(\{g_{j,z}, g_{j,r}, \{g_{j,i}\}_{i=1}^{n}\}_{j=1}^{k}\Big).$$

Sign($\mathsf{sk}, (M_1, \ldots, M_n)$): to sign $(M_1, \ldots, M_n) \in \mathbb{G}^n$ using $\mathsf{sk} = \big(\{\chi_i, \{\gamma_{j,i}\}_{j=1}^k\}_{i=1}^n\big)$, compute and output $\sigma = (z, r_1, \ldots, r_k) \in \mathbb{G}^{k+1}$, where

$$z = \prod_{i=1}^{n} M_i^{-\chi_i},$$

$$r_j = \prod_{i=1}^{n} M_i^{-\gamma_{j,i}} \qquad\qquad j \in \{1, \ldots, k\}.$$

SignDerive($\mathsf{pk}, \{(\omega_i, \sigma^{(i)})\}_{i=1}^{\ell}$): given a public key pk and ℓ tuples $(\omega_i, \sigma^{(i)})$, where $\omega_i \in \mathbb{Z}_p$ for each i, parse $\sigma^{(i)}$ as $\sigma^{(i)} = \big(z_i, r_{i,1}, \ldots, r_{i,k}\big) \in \mathbb{G}^{k+1}$ for $i = 1$ to ℓ. Then, compute and return $\sigma = (z, r_1, \ldots, r_k)$, where $z = \prod_{i=1}^{\ell} z_i^{\omega_i}$, $r_j = \prod_{i=1}^{\ell} r_{i,j}^{\omega_i}$ for $j = 1$ to k.

Verify($\mathsf{pk}, \sigma, (M_1, \ldots, M_n)$): given $\sigma = (z, r_1, \ldots, r_k) \in \mathbb{G}^{k+1}$ and $(M_1, \ldots, M_n)$, return 1 if and only if $(M_1, \ldots, M_n) \neq (1_{\mathbb{G}}, \ldots, 1_{\mathbb{G}})$ and, for each $j \in \{1, \ldots, k\}$, the following equality holds:

$$1_{\mathbb{G}_T} = e(g_{j,z}, z) \cdot e(g_{j,r}, r_j) \cdot \prod_{i=1}^{n} e(g_{j,i}, M_i). \tag{6}$$

In order to adapt the unbounded simulation-sound proof system of Section 3, we need to commit to the components of $(z, r_1, \ldots, r_k)$ and NIWI arguments showing that committed group elements satisfy the pairing product equations (6). Under the k-linear assumption, committing to a group element requires $k+1$ group elements (see, *e.g.*, [15] for details) whereas each equation of the form (6) costs $k+1$ elements to prove. Overall, we thus need $(k+1)(2k+1)$ group elements and a one-time verification key pair (SVK, σ).

In the relatively-sound QA-NIZK proof of Section 4, the proof element π_0 remains unchanged and we simply need to replace the triple (z, r, u) by a one-time linearly homomorphic signature $(z, r_1, \ldots, r_k)$. Hence, we only need $k+2$ group elements.

E Comparisons

This section compares the various NIZK proofs of linear subspace membership based on the DLIN assumption. Comparisons are given in terms of CRS size, proof size, the number of pairing evaluations for the verifier and the need for a computational assumption to prove the soundness property.

In the table, we consider our basic proof system (without any form of simulation-soundness, where each proof is a one-time linearly homomorphic signature (z, r, u)), its unbounded simulation-sound variant and the relatively simulation-sound variant of Section 4. We compare these with the original Groth-Sahai proofs, their most efficient unbounded simulation-sound extensions due to Camenisch *et al.* [15] and the Jutla-Roy techniques [38] with and without relative soundness.

Table 1. Comparison between proof systems for linear subspaces

Proof systems	CRS size$^{\diamondsuit}$ *	Proof length$^{\diamondsuit}$	# of pairings† at verification	Soundness property
Groth-Sahai [33]	6	$3t + 2n$	$3n(t+3)$	perfect
Jutla-Roy [38]	$4t(n-t)+3$	$2(n-t)$	2(n-t)(t+2)	computational
Jutla-Roy RSS [38] + [37]	$4t(n+1-t)+3$	$2(n+1-t)+1$	$2(n+1-t)(t+2)$	computational
Groth-Sahai USS [15]	18	$6t+2n+52^{\ddagger}$	$O(tn)$	computational
Our basic QA-NIZK proofs	$2n+3t+4$	3	$2n+2$	computational
Our RSS QA-NIZK proofs	$4n+8t+6$	4	$2n+6$	computational
Our USS QA-NIZK proofs	$2n+3t+3L+10$	$20^{\ddagger}$	$2n+30$	computational

n: number of equations; t: number of variables; L: length of a hashed one-time verification key

$\diamondsuit$ These sizes are measured in terms of number of group elements.
$*$ The description $\rho \in \mathbb{G}^{t \times n}$ of the language is not counted as being part of the CRS here.
† The table does not consider optimizations using randomized batch verification techniques here.
‡ We consider instantiations using Groth's one-time signature [32], where verification keys and signatures consist of 3 group elements and two elements of $\mathbb{Z}_p$, respectively.

As can be observed in the table, our constructions all yield constant-size arguments. Moreover, the number of pairing evaluations is always independent of the number of variables t, which substantially fastens the verification process when $t \approx n/2$.

We also note that randomized batch verification techniques can be used to drastically reduce the number of pairing computations. In our USS system, for example, the number of pairings drops to $n+18$ if the two verification equations are processed together and further optimizations are possible.

Our common reference strings always fit within $O(t+n)$ group elements (with another $O(L)$ elements in the USS variant) and thus provide significant savings w.r.t. [38] when $t \approx n/2$.

F Proof of Theorem 1

Proof. The quasi-adaptive completeness property follows directly from the correctness of the randomizable linearly homomorphic signature of Section 2.5. We thus focus on the quasi-adaptive zero-knowledge and quasi-adaptive unbounded simulation-soundness properties.

Quasi-Adaptive Zero-Knowledge. To prove this property we consider a sequence of two games which begins with a game where the adversary has access to a real prover P on a real CRS ψ. In the second game, the adversary will be faced with a simulator $(\mathsf{S}_1, \mathsf{S}_2)$.

Game_1**:** is a game where the adversary $\mathcal{A}$ is given the description of the language $\mathcal{L}_\rho$ and is granted access to a real CRS ψ and an actual prover $\mathsf{P}(\psi, ., .)$ which takes as input a vector $\boldsymbol{v}$ along with a witness $\boldsymbol{x} \in \mathbb{Z}_p^t$ such that $\boldsymbol{v} = g^{\boldsymbol{x}\cdot\mathbf{A}}$. At each invocation, the oracle outputs a genuine proof π by running the legal P algorithm. The adversary is allowed to query $\mathsf{P}(\psi, ., .)$ a polynomial number of times and eventually outputs a bit $\beta \in \{0,1\}$. We denote by S_1 the event that $\beta = 1$.

Game_2**:** This game is identical to Game_2 with the difference that, when the $\mathsf{P}(\psi, ., .)$ oracle is queried on a pair $(\boldsymbol{v}, \boldsymbol{x})$, it does not use the witness $\boldsymbol{x} \in \mathbb{Z}_p^t$ anymore at step 1 of the proving algorithm. Instead, it uses the the private key $\mathsf{sk}_{rand} = \{(\chi_i, \gamma_i, \delta_i)\}_{i=1}^n$ to compute a one-time signature $(z, r, u) = \big(\prod_{j=1}^n v_j^{-\chi_j}, \prod_{j=1}^n v_j^{-\gamma_j}, \prod_{j=1}^n v_j^{-\delta_j}\big)$ on the vector $\boldsymbol{v} = (v_1, \ldots, v_n) \in \mathbb{G}^n$. The remaining parts of π are generated as in the real $\mathsf{P}(\psi, ., .)$ oracle in steps 2 and 3 of the proof generation algorithm. Although, the witness $\boldsymbol{x} \in \mathbb{Z}_p^t$ is never used, it is easy to see that (z, r, u) has exactly the same distribution as in Game_2 if $\boldsymbol{v} \in \mathcal{L}_\rho$ (*i.e.*, as long as $\boldsymbol{v} = g^{\boldsymbol{x}\cdot\mathbf{A}}$ for some $\boldsymbol{x} \in \mathbb{Z}_p^t$). We thus have $\Pr[S_2] = \Pr[S_1]$.

We define the simulator $(\mathsf{S}_1, \mathsf{S}_2)$ by having S_1 generate the CRS ψ as in Game_1 (so that ψ has the same distribution as the real CRS) and letting S_2 generate proofs without using the witnesses as in Game_2. It easily comes that the system is perfectly quasi-adaptive zero-knowledge as, for all language members $\boldsymbol{v} \in \mathbb{G}^n$, simulated proofs are distributed as real proofs.

Quasi-Adaptive Unbounded Simulation-Soundness. To prove this property, we proceed again with a sequence of games.

Game_1**:** is the real game where the adversary $\mathcal{A}$ is given the description of the language $\mathcal{L}_\rho$ and is granted access to a simulated CRS ψ and a simulated prover $\mathsf{S}_2(\psi, \tau_{sim}, ., .)$ which takes as input a vector-label pair $(\boldsymbol{v}, \mathsf{lbl})$ and returns a simulated proof π that $\boldsymbol{v} \in \mathcal{L}_\rho$. To generate $\rho \in \mathbb{G}^{t\times n}$ according to the distribution D_Γ, the challenger generates a matrix $\mathbf{A} \in \mathbb{Z}_p^{t\times n}$ with the appropriate distribution (recall that D_Γ is efficiently samplable by hypothesis) and computes $\rho = g^{\mathbf{A}}$. In addition, the challenger $\mathcal{B}$ computes a basis $\mathbf{W} \in \mathbb{Z}_p^{n\times(n-t)}$ of the right kernel of $\mathbf{A}$ and retains it for later use. The adversary is allowed to query the simulated prover $\mathsf{S}_2(\psi, \tau_{sim}, ., .)$ on polynomially many occasions. The game ends with the adversary $\mathcal{A}$ outputting an element $\boldsymbol{v}^\star$, a proof $\pi^\star$ and a label $\mathsf{lbl}^\star$. The adversary is deemed successful if $\boldsymbol{v}^\star \notin \mathcal{L}_\rho$ (*i.e.*, $\boldsymbol{v}^\star$ is not in the row space of

$\rho \in \mathbb{G}^{t\times n}$) but $(\pi^\star, \mathsf{lbl}^\star)$ is a valid proof. We denote by S_1 the latter event, which $\mathcal{B}$ can recognize by testing if $\boldsymbol{v} = (v_1, \ldots, v_n) \in \mathbb{G}^n$ satisfies $\prod_{j=1}^n v_j^{w_{ji}} = 1_\mathbb{G}$ for each column $\boldsymbol{w}_i^\top = (w_{1i}, \ldots, w_{ni})^\top$ of $\mathbf{W}$. Indeed, the vectors $\mathbf{y} \in \mathbb{Z}_p^n$ in the row space of $\mathbf{A}$ are exactly those for which $\mathbf{y} \cdot \mathbf{W} = \mathbf{0}$.

Game_2**:** This game is identical to Game_1 but the challenger $\mathcal{B}$ aborts if the adversary $\mathcal{A}$ outputs a fake proof $\pi^\star$ that recycles one of the one-time verification keys appearing in outputs of the $\mathsf{S}_2(\psi, \tau_{sim}, ., .)$ oracle. Clearly, Game_2 and Game_1 are identical until the latter event occurs and this event contradicts the strong unforgeability of Σ: if q denotes the number of queries to $\mathsf{S}_2(\psi, \tau_{sim}, ., .)$, a standard argument shows that $|\Pr[S_2] - \Pr[S_1]| \leq q \cdot \mathbf{Adv}^{\text{suf-ots}}(\mathcal{B})$.

Game_3**:** This game is identical to Game_2 but we modify the generation of pk_{rand} when the public key is set up. Namely, the vectors $(\boldsymbol{f}_1, \boldsymbol{f}_2, \{\boldsymbol{f}_{3,i}\}_{i=0}^L)$ are chosen by setting $\boldsymbol{f}_1 = (f_1, 1_\mathbb{G}, g)$ and $\boldsymbol{f}_2 = (1_\mathbb{G}, f_2, g)$ where $f_1, f_2 \xleftarrow{R} \mathbb{G}$ are chosen at random. As for $\{\boldsymbol{f}_{3,i}\}_{i=0}^L$, they are obtained as

$$\begin{aligned} \boldsymbol{f}_{3,0} &= \boldsymbol{f}_1^{\xi_{0,1}} \cdot \boldsymbol{f}_2^{\xi_{0,2}} \cdot (1,1,g)^{\xi_{0,3}} \cdot (1,1,g)^{\mu\cdot\zeta-\rho_0} \\ \boldsymbol{f}_{3,i} &= \boldsymbol{f}_1^{\xi_{i,1}} \cdot \boldsymbol{f}_2^{\xi_{i,2}} \cdot (1,1,g)^{\xi_{i,3}} \cdot (1,1,g)^{-\rho_i}, \qquad\qquad i \in \{1,\ldots,L\} \end{aligned} \tag{7}$$

with $\mu \xleftarrow{R} \{0,\ldots,L\}$, $\xi_{0,1}, \xi_{1,1}, \ldots, \xi_{L,1} \xleftarrow{R} \mathbb{Z}_p$, $\xi_{0,2}, \xi_{1,2}, \ldots, \xi_{L,2} \xleftarrow{R} \mathbb{Z}_p$, $\xi_{0,3}, \xi_{1,3}, \ldots, \xi_{L,3} \xleftarrow{R} \mathbb{Z}_p$ and $\rho_0, \rho_1, \ldots, \rho_L \xleftarrow{R} \{0,\ldots,\zeta-1\}$, with $\zeta = 2(q+1)$ and where q is the number of queries to the $\mathsf{S}_2(\psi, \tau_{sim}, .)$ oracle. Note that this change is only conceptual since the distribution of $\{\boldsymbol{f}_{3,i}\}_{i=0}^L$ has not changed since Game_2. We thus have $\Pr[S_3] = \Pr[S_2]$.

Game_4**:** This game is like Game_3 but we consider an event Good which causes the challenger $\mathcal{B}$ to abort if it does *not* occur. Let $\mathsf{SVK}_1, \ldots, \mathsf{SVK}_q$ be the distinct one-time verification keys appearing in outputs of the S_2 oracle throughout the game. Let also $\mathsf{SVK}^\star$ be the verification key involved in the fake proof $\pi^\star$ produced by $\mathcal{A}$. We know that $\mathsf{SVK}^\star \notin \{\mathsf{SVK}_1, \ldots, \mathsf{SVK}_q\}$ unless the failure event introduced in Game_2 occurs. For each verification key $\mathsf{SVK} \in \{0,1\}^L$, we consider the function $J(\mathsf{SVK}) = \mu \cdot \zeta - \rho_0 - \sum_{i=1}^L \rho_i \cdot \mathsf{SVK}[i]$, where $\{\rho_i\}_{i=0}^L$ are the values internally defined by the simulator in Game_3. We also define Good to be the event that

$$J(\mathsf{SVK}^\star) = 0 \qquad \wedge \qquad \bigwedge_{j\in\{1,\ldots,q\}} J(\mathsf{SVK}_j) \neq 0. \tag{8}$$

We remark that the random exponents $\rho_0, \rho_1, \ldots, \rho_L$ are chosen independently of $\mathcal{A}$'s view: this means that the simulator could equivalently define $\{\boldsymbol{f}_{3,i}\}_{i=0}^L$ first and only choose $\{\rho_i\}_{i=0}^L$ – together with values $\{\xi_{3,i}\}_{i=0}^L$ explaining the $\{\boldsymbol{f}_{3,i}\}_{i=0}^L$ – at the very end of the game, when $\mathsf{SVK}^\star, \mathsf{SVK}_1, \ldots, \mathsf{SVK}_q, \mathsf{SVK}$ have been defined. The same analysis as [54] (using the simplifications of Bellare and Ristenpart [6, Theorem 3.1]) shows that $\Pr[S_4 \wedge \mathsf{Good}] \geq \Pr[S_3]^2/(27\cdot(q+1)\cdot(L+1))$.

This follows from the fact that, for any set of queries, a lower bound on the probability of event Good is $1/(2q(L+1))$. Indeed, from the known results [54, 35] on the programmability of Waters' hash function, we know that the probability, taken over the choice of $(\mu, \rho_0, \ldots, \rho_L)$, to meet the conditions (8) is at least $1/(2q(L+1))$.

Game_5**:** We modify again the way to compute pk_{rand} in the generation of the public key. Namely, the vectors $\boldsymbol{f_1} = (f_1, 1_\mathbb{G}, g)$, $\boldsymbol{f_2} = (1_\mathbb{G}, f_2, g)$ are chosen as before. However, instead of generating $\{\boldsymbol{f}_{3,i}\}_{i=0}^L$ as in Game_4, we set them as

$$\begin{aligned} \boldsymbol{f}_{3,0} &= \boldsymbol{f}_1^{\xi_{0,1}} \cdot \boldsymbol{f}_2^{\xi_{0,2}} \cdot (1,1,g)^{\mu\cdot\zeta-\rho_0} \\ \boldsymbol{f}_{3,i} &= \boldsymbol{f}_1^{\xi_{i,1}} \cdot \boldsymbol{f}_2^{\xi_{i,2}} \cdot (1,1,g)^{-\rho_i}, \qquad\qquad i \in \{1,\ldots,L\} \end{aligned} \tag{9}$$

which amounts to setting $\xi_{0,3} = \xi_{1,3} = \ldots = \xi_{L,3} = 0$. Clearly, $\{\boldsymbol{f}_{3,i}\}_{i=0}^L$ are no longer uniform in the span of $\big(\boldsymbol{f}_1, \boldsymbol{f}_2, (1,1,g)\big)$. Still, this change should not be noticeable to $\mathcal{A}$ if the DLIN assumption holds in $\mathbb{G}$. Concretely, if the adversary wins (recall that the challenger can still detect

this event using the matrix $\mathbf{W} \in \mathbb{Z}_p^{n\times(n-t)}$ as explained in Game_1) with substantially different probabilities in Game_5 and Game_4, we can construct a DLIN distinguisher $\mathcal{B}^{\mathrm{DLIN}}$ in the group $\mathbb{G}$. This distinguisher uses the random self-reducibility of DLIN to construct many independent-looking instances from the same distribution out of a given instance. The distinguisher then runs $\mathcal{A}$ on input of the CRS that was generated using the DLIN instances and it eventually outputs 1 if the adversary wins. We can thus write $|\Pr[S_5 \wedge \mathsf{Good}] - \Pr[S_4 \wedge \mathsf{Good}]| \leq \mathbf{Adv}_{\mathcal{B}^{\mathrm{DLIN}}}(\lambda)$.

In Game_5, we show that a successful adversary implies an algorithm $\mathcal{B}$ solving a given SDP instance (g_z, g_r, h_z, h_u) with non-negligible probability, which *a fortiori* breaks the DLIN assumption in $\mathbb{G}$.

By hypothesis, we know that $\mathcal{A}$ manages to create a proof $\pi^\star = (\mathsf{SVK}^\star, \boldsymbol{C}_z^\star, \boldsymbol{C}_r^\star, \boldsymbol{C}_u^\star, \boldsymbol{\pi}_1^\star, \boldsymbol{\pi}_2^\star, \sigma^\star)$ for a vector $\boldsymbol{v}^\star = (v_1^\star, \ldots, v_n^\star)$ outside the row space of ρ but $(\boldsymbol{C}_z^\star, \boldsymbol{C}_r^\star, \boldsymbol{C}_u^\star, \boldsymbol{\pi}_1^\star, \boldsymbol{\pi}_2^\star) \in \mathbb{G}^{15}$ and $\sigma^\star$ satisfy the verification equations. At this point, if the event Good introduced in Game_4 occurs, we must have $J(\mathsf{SVK}^\star) = 0$, which implies that $\boldsymbol{f}_{\mathsf{SVK}^\star} = \boldsymbol{f}_{3,0} \cdot \prod_{i=1}^{L+1} \boldsymbol{f}_{3,i}^{\mathsf{SVK}^\star[i]}$ lies in $\mathrm{span}(\boldsymbol{f}_1, \boldsymbol{f}_2)$. Consequently, $\boldsymbol{C}_z^\star$, $\boldsymbol{C}_r^\star$ and $\boldsymbol{C}_u^\star$ are necessarily perfectly binding and extractable commitments. Using $(\log_g(f_1), \log_g(f_2))$, algorithm $\mathcal{B}$ can thus extract the committed group elements $(z^\star, r^\star, u^\star) \in \mathbb{G}^3$ by BBS-decrypting the ciphertexts $(\boldsymbol{C}_z^\star, \boldsymbol{C}_r^\star, \boldsymbol{C}_u^\star)$. Since $(\boldsymbol{\pi}_1^\star, \boldsymbol{\pi}_2^\star)$ are perfectly sound Groth-Sahai proofs, the extracted elements $(z^\star, r^\star, u^\star)$ necessarily satisfy

$$1_{\mathbb{G}_T} = e(g_z, z^\star) \cdot e(g_r, r^\star) \cdot \prod_{i=1}^{n} e(g_i, v_i^\star) = e(h_z, z^\star) \cdot e(h_u, u^\star) \cdot \prod_{i=1}^{n} e(h_i, v_i^\star). \tag{10}$$

Having extracted $(z^\star, r^\star, u^\star)$, $\mathcal{B}$ also computes

$$z^\dagger = \prod_{i=1}^{n} v_i^{\star-\chi_i} \qquad r^\dagger = \prod_{i=1}^{n} v_i^{\star-\gamma_i} \qquad u^\dagger = \prod_{i=1}^{n} v_i^{\star-\delta_i}, \tag{11}$$

so that $(z^\dagger, r^\dagger, u^\dagger)$ also satisfies (10). Since $(z^\dagger, r^\dagger, u^\dagger)$ and $(z^\star, r^\star, u^\star)$ both satisfy (10), the triple

$$(z^\ddagger, r^\ddagger, u^\ddagger) = \left(\frac{z^\star}{z^\dagger}, \frac{r^\star}{r^\dagger}, \frac{u^\star}{u^\dagger}\right)$$

necessarily satisfies $e(g_z, z^\ddagger) \cdot e(g_r, r^\ddagger) = e(h_z, z^\ddagger) \cdot e(h_u, u^\ddagger) = 1_{\mathbb{G}_T}$. To conclude the proof, we argue that $z^\ddagger \neq 1_{\mathbb{G}}$ with overwhelming probability.

To do this, we observe that, if the event Good defined in Game_4 actually comes about, then $\mathcal{B}$ never leaks any more information about $(\chi_1, \ldots, \chi_n)$ than $\mathcal{A}$ can infer by just observing $\{(z_j, r_j, u_j)\}_{j=1}^t$ in the public key. Indeed, in this case we have $J(\mathsf{SVK}^\star) = 0$ and $J(\mathsf{SVK}_j) \neq 0$ for each $j \in \{1, \ldots, q\}$. This means that, in the simulated proofs returned by $\mathsf{S}_2(\psi, \tau_{sim}, ., .)$, the proofs $(\boldsymbol{\pi_1}, \boldsymbol{\pi_2})$ are perfectly witness indistinguishable as they are generated for a perfectly hiding Groth-Sahai CRS. For these simulated proofs, the built-in homomorphic signatures $(\boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi_2})$ leak nothing about the specific vector $(\chi_1, \ldots, \chi_n)$ used by $\mathcal{B}$. As a consequence, the same arguments as in [43, Theorem 1] show that $z^\dagger \neq z^\star$ with probability $1 - 1/p$. Specifically, in the CRS, $\{(g_i, h_i)\}_{i=1}^n$ and $\{(z_i, r_i, u_i)\}_{i=1}^t$ provide the adversary with a system of $2n + t < 3n$ equations in $3n$ unknowns $\{(\chi_i, \gamma_i, \delta_i)\}_{i=1}^n$, which leaves $z^\dagger$ completely undetermined as long as $\boldsymbol{v}^\star$ is linearly independent of the rows of $\big(G_{i,j}\big)_{i,j}$. We thus find

$$\Pr[S_5 \wedge \mathsf{Good}] = \mathbf{Adv}_{\mathcal{B}}^{\mathrm{SDP}}(\lambda) \cdot \left(1 - \frac{1}{p}\right)^{-1}.$$

In turn, in Game_5, $\mathcal{B}$ implies a PPT distinguisher $\mathcal{B}^{\mathrm{DLIN}'}$ for the DLIN assumption such that we have the inequality $\Pr[S_5 \wedge \mathsf{Good}] < \frac{1}{2} \cdot \mathbf{Adv}_{\mathcal{B}^{\mathrm{DLIN}'}}(\lambda) \cdot \left(1 - \frac{1}{p}\right)^{-1}$. If $\mathbf{Adv}^{\mathrm{DLIN}}(\lambda)$ denotes the maximal advantage of any PPT distinguisher against the DLIN assumption in $\mathbb{G}$, the probability of event

$S_4 \wedge \mathsf{Good}$ can be bounded as $\Pr[S_4 \wedge \mathsf{Good}] \leq \frac{3}{2} \cdot \mathbf{Adv}^{\mathrm{DLIN}}(\lambda) \cdot \big(1 - \frac{1}{p}\big)^{-1}$ in Game_5 in Game_4. This in turn yields

$$\Pr[S_3] \leq 7 \cdot \sqrt{q \cdot (L+1) \cdot \mathbf{Adv}^{\mathrm{DLIN}}(\lambda) \cdot \big(1 - \frac{1}{p}\big)^{-1}},$$

so that $\mathcal{A}$'s advantage in breaking the unbounded simulation-soundness of the system is at most

$$\mathbf{Adv}^{\mathrm{uss}}(\mathcal{A}) \leq \mathbf{Adv}^{\text{suf-ots}}(\lambda) + 7 \cdot \sqrt{q \cdot (L+1) \cdot \mathbf{Adv}^{\mathrm{DLIN}}(\lambda) \cdot \big(1 - \frac{1}{p}\big)^{-1}}. \tag{12}$$

□

G Proof of Theorem 2

Proof. The completeness property is straightforward to verify. To establish the result, we separately prove the relative quasi-adaptive zero-knowledge and relative quasi-adaptive simulation-soundness properties.

Quasi-Adaptive Relative Zero-Knowledge. We consider a sequence of two games which begins with a game where the adversary has oracle access to the actual prover P and a public verifier on a real CRS ψ. In the last game, the adversary will be interacting with a simulator $(\mathsf{S}_1, \mathsf{S}_2)$ and the private verifier.

Game_1**:** is a game where the adversary $\mathcal{A}$ is given the description ρ of the language $\mathcal{L}_\rho$ and a real CRS $\psi = (\mathbf{CRS}_1, \mathbf{CRS}_2)$. In addition, the adversary has access to a public verification oracle $\mathsf{V}(\psi, ., .)$, even though it can run the verification algorithm by itself. This will be useful to show that the private verifier always agrees with the public one when it interacts with a PPT adversary. At some point, the adversary chooses a pair $(\boldsymbol{v}, \mathsf{lbl})$ along with a witness $\boldsymbol{x} \in \mathbb{Z}_p^t$ such that $\boldsymbol{v} = g^{\boldsymbol{x} \cdot \mathbf{A}}$. The challenger replies by returning an actual proof π produced by running $\mathsf{P}(\psi, \boldsymbol{v}, \boldsymbol{x}, \mathsf{lbl})$. When $\mathcal{A}$ terminates, it outputs a bit $\beta \in \{0, 1\}$. We denote by S_1 the event that $\beta = 1$.

Game_2**:** is like Game_1 but the adversary's public verification oracle $\mathsf{V}(\psi, ., .)$ is replaced by the private verification oracle $\mathsf{W}(\psi, \tau_v, ., .)$. Since the private verification algorithm begins by running the public one, both games are clearly identical until $\mathcal{A}$ queries the verification oracle on input of a candidate $(\boldsymbol{v}, (z, r, u, \pi_0), \mathsf{lbl})$ that is accepted by $\mathsf{V}(\psi, ., .)$ but rejected by $\mathsf{W}(\psi, \tau_v, ., .)$. If we call this event F_2, we have $|\Pr[S_1] - \Pr[S_2]| \leq \Pr[F_2]$.

Claim 1. The probability of event F_2 can be bounded as $\Pr[F_2] \leq \mathbf{Adv}^{\mathrm{SDP}}(\lambda) + \frac{1}{p}$.

Proof. We first remark that event F_2 can only occur for a candidate $\big(\boldsymbol{v}, (z, r, u, \pi_0), \mathsf{lbl}\big)$ such that the vector $(v_1, \ldots, v_n, \pi_0, v_1^\alpha, \ldots, v_n^\alpha)$ is not in the span of $\{\boldsymbol{H}_{2i-1}, \boldsymbol{H}_{2i}\}_{i=1}^t$. Indeed, otherwise, there would exist $\boldsymbol{x} = (x_1, \ldots, x_t) \in \mathbb{Z}_p^t$ such that $(v_1, \ldots, v_n) = g^{\boldsymbol{x} \cdot \mathbf{A}}$ and $\pi_0 = \prod_{i=1}^t (W_i^\alpha Y_i)^{x_i}$. In this case, we would also have

$$g^{\boldsymbol{x} \cdot \mathbf{A} \cdot (\boldsymbol{e}^\top + \alpha \cdot \boldsymbol{d}^\top)} = \prod_{i=1}^t (W_i^\alpha Y_i)^{x_i},$$

and the private verifier $\mathsf{W}(\psi, \tau_v, ., .)$ would accept the proof (z, r, u, π_0).

It comes that the only way for the adversary to cause a divergence between $\mathsf{W}(\psi, \tau_v, ., .)$ and $\mathsf{V}(\psi, ., .)$ is to create a valid-looking one-time linearly homomorphic signature (z, r, u) on a vector outside $\mathrm{span}\big(\{\boldsymbol{H}_{2i-1}, \boldsymbol{H}_{2i}\}_{i=1}^t\big)$. The result of [43][Theorem 1] shows that this occurs with probability at most $\Pr[F_2] \leq \mathbf{Adv}^{\mathrm{SDP}}(\lambda) + \frac{1}{p}$. ■

Game_3**:** This game like Game_2 but, when the adversary outputs its triple $(\boldsymbol{v}, \boldsymbol{x}, \mathsf{lbl})$, the challenger does not use the witness $\boldsymbol{x} \in \mathbb{Z}_p^t$ any longer. To simulate the proof for $(\boldsymbol{v}, \boldsymbol{x}, \mathsf{lbl})$, it first computes $\alpha = H(\rho, \boldsymbol{v}, \mathsf{lbl})$. Then, using the private vectors $\boldsymbol{d}, \boldsymbol{e} \in \mathbb{Z}_p^n$, it computes $\pi_0 = \prod_{j=1}^n v_j^{e_j + \alpha \cdot d_j}$ before using the private key $\mathsf{sk}_{ots} = \{(\chi_i, \gamma_i, \delta_i)\}_{i=1}^{2n+1}$ to compute a one-time signature

$$(z, r, u) = \Big(\prod_{j=1}^{2n+1} v_j^{-\chi_j}, \prod_{j=1}^{2n+1} v_j^{-\gamma_j}, \prod_{j=1}^{2n+1} v_j^{-\delta_j} \Big)$$

on the vector $\tilde{\boldsymbol{v}} = (v_1, \ldots, v_{2n+1})$, where $v_{n+1} = \pi_0$ and $v_{n+i+1} = v_i^{\alpha}$ for $i = 1$ to n. The resulting proof is easily seen to have the same distribution as in Game_2 when $\boldsymbol{v} \in \mathcal{L}_\rho$. We thus have $\Pr[S_3] = \Pr[S_2]$.

We define the simulator $(\mathsf{S}_1, \mathsf{S}_2)$ by having S_1 generate the CRS ψ as in Game_3 (observe that it has not changed since Game_1) and S_2 generate compute the proof without using the witnesses as in Game_3. The verification oracle is implemented as in Game_2 and Game_3. It easily comes that the system is computationally quasi-adaptive relatively zero-knowledge if the SDP assumption holds.

Quasi-Adaptive Relative Simulation-Soundness. We have to prove that, even if the simulator $(\mathsf{S}_1, \mathsf{S}_2)$ provides the adversary $\mathcal{A}$ with a simulated proof π for a pair $(\boldsymbol{v}, \mathsf{lbl})$, where $\boldsymbol{v} \in \mathbb{G}^n$ may not be in $\mathcal{L}_\rho$, $\mathcal{A}$ will remain unable to produce a new proof $(\boldsymbol{v}^\star, \pi^\star, \mathsf{lbl}^\star) \neq (\boldsymbol{v}, \pi, \mathsf{lbl})$ such that $\boldsymbol{v}^\star \notin \mathcal{L}_\rho$.

To prove the result, we rely on the smoothness of the projective hash function and on a specific property of the one-time linearly homomorphic signature of Section 2.5: namely, unless the SDP assumption is false, it is hard to compute two distinct signatures on the same vector, even when the private key is available.

We thus proceed with a sequence of games where the first game is the actual game and the last one is a game where the adversary has statistically no advantage. In Game_i, we denote by S_i the event that the adversary wins.

Game_1**:** is the real game where the adversary $\mathcal{A}$ is given the description of $\mathcal{L}_\rho$, a simulated CRS ψ and access to a simulated prover $\mathsf{S}_2(\psi, \tau_{sim}, ., .)$ which is queried only once. On this occasion, S_2 takes as input a vector $\boldsymbol{v}^\dagger \in \mathbb{G}^n$ and a label $\mathsf{lbl}^\dagger$ and it produces a simulated proof $\pi^\dagger = (z^\dagger, r^\dagger, u^\dagger, \pi_0^\dagger)$ that $\boldsymbol{v}^\dagger \in \mathcal{L}_\rho$. To generate $\rho \in \mathbb{G}^{t \times n}$ according to the distribution D_Γ at the beginning of the game, the challenger computes $\rho = g^{\mathbf{A}}$ after having sampled a matrix $\mathbf{A} \in \mathbb{Z}_p^{t \times n}$ with the appropriate distribution (which is possible since D_Γ is efficiently samplable). Moreover, the challenger $\mathcal{B}$ computes a basis $\mathbf{W} \in \mathbb{Z}_p^{n \times (n-t)}$ of the right kernel of $\mathbf{A}$. The adversary is allowed to query $\mathsf{S}_2(\psi, \tau_{sim}, ., .)$ exactly once and the private verifier $\mathsf{W}(\psi, \tau_v, ., .)$ on polynomially many occasions. When $\mathcal{A}$ terminates, it outputs a triple $(\boldsymbol{v}^\star, \pi^\star, \mathsf{lbl}^\star)$. The adversary is successful if $\boldsymbol{v}^\star \notin \mathcal{L}_\rho$ but $(\pi^\star, \mathsf{lbl}^\star)$ passes the private verification test and $(\boldsymbol{v}^\star, \pi^\star, \mathsf{lbl}^\star) \neq (\boldsymbol{v}^\dagger, \pi^\dagger, \mathsf{lbl}^\dagger)$. We denote by S_1 the latter event. Note that $\mathcal{B}$ can recognize this event as it can test if $\boldsymbol{v}^\star \notin \mathcal{L}_\rho$ by checking whether $\boldsymbol{v}^\star = (v_1^\star, \ldots, v_n^\star) \in \mathbb{G}^n$ satisfies $\prod_{j=1}^n v_j^{\star w_{ji}} = 1_{\mathbb{G}}$ for each column $\boldsymbol{w}_i^\top = (w_{1i}, \ldots, w_{ni})^\top$ of $\mathbf{W}$. Indeed, the vectors $\mathbf{y} \in \mathbb{Z}_p^n$ in the row space of $\mathbf{A}$ are exactly those for which $\mathbf{y} \cdot \mathbf{W} = \mathbf{0}$.

Game_2**:** In this game, we modify the behavior of the private verification oracle $\mathsf{W}(\psi, \tau_v, ., .)$. At each invocation (including the final invocation on the adversary's output $(\boldsymbol{v}^\star, \pi^\star, \mathsf{lbl}^\star)$) on input of a triple $(\boldsymbol{v}, \pi, \mathsf{lbl})$, the modified private verification oracle outputs 0 if $(\boldsymbol{v}, \pi, \mathsf{lbl}) \neq (\boldsymbol{v}^\dagger, \pi^\dagger, \mathsf{lbl}^\dagger)$ but $H(\rho, \boldsymbol{v}, \mathsf{lbl}) = H(\rho, \boldsymbol{v}^\dagger, \mathsf{lbl}^\dagger)$. Clearly, Game_1 and Game_2 proceed identically until the latter event, called F_2, occurs. We have $|\Pr[S_2] - \Pr[S_1]| \leq \Pr[F_2]$. Moreover, F_2 is unlikely to occur if H is a collision-resistant hash function: we have $\Pr[F_2] \leq \mathbf{Adv}^{\mathrm{CR}}(\lambda)$.

Game_3**:** This game is identical to Game_2 with the following difference. When the private verification oracle $\mathsf{W}(\psi, \tau_v, ., .)$ is run on input of the adversary's proof $(\boldsymbol{v}^\star, \pi^\star, \mathsf{lbl}^\star)$, it returns 0 in the event

that $\pi^\star = (z^\star, r^\star, u^\star, \pi_0^\star)$ is such that $(z^\star, r^\star, u^\star) \neq (z^\dagger, r^\dagger, u^\dagger)$ but $(\boldsymbol{v}^\star, \mathsf{lbl}^\star) = (\boldsymbol{v}^\dagger, \mathsf{lbl}^\dagger)$. If we call F_3 the event that the private verification oracle $\mathsf{W}(\psi, \tau_v, ., .)$ rejects a proof that would have been accepted in Game_2, we have $|\Pr[S_3] - \Pr[S_2]| \leq \Pr[F_3]$. Moreover, F_3 implies a breach in the SDP assumption. Indeed, if $\pi_0^\star \neq \pi_0^\dagger$, $\mathsf{W}(\psi, \tau_v, ., .)$ would not accept $\pi^\star$ in Game_2 either, regardless of whether $(z^\star, r^\star, u^\star) \neq (z^\dagger, r^\dagger, u^\dagger)$ or not. If $\pi_0^\star = \pi_0^\dagger$, event F_3 provides the challenger with two distinct linearly homomorphic signatures $(z^\star, r^\star, u^\star)$ and $(z^\dagger, r^\dagger, u^\dagger)$ on the same vector $(v_1^\star, \ldots, v_n^\star, \pi_0^\star, v_1^{\star\alpha^\star}, \ldots, v_n^{\star\alpha^\star})$ as we also have $\alpha^\star = \alpha^\dagger$. As mentioned in Section 2.5 (and as can be easily observed from the verification equations), this would contradict the SDP assumption and we thus have $|\Pr[S_3] - \Pr[S_2]| \leq \mathbf{Adv}^{\mathrm{SDP}}(\lambda)$.

Game_4**:** In this game, we further modify the behavior of $\mathsf{W}(\psi, \tau_v, ., .)$ when it assesses the the adversary's output $(\boldsymbol{v}^\star, \pi^\star, \mathsf{lbl}^\star)$. Using the basis $\mathbf{W}$ of the right kernel of $\mathbf{A}$, the challenger $\mathcal{B}$ first checks if $\boldsymbol{v}^\star \notin \mathcal{L}_\rho$ and forces $\mathsf{W}(\psi, \tau_v, ., .)$ to return 0 whenever this is the case. If we denote by F_4 the event that $\mathsf{W}(\psi, \tau_v, ., .)$ rejects an adversarially-generated triple $(\boldsymbol{v}^\star, \pi^\star, \mathsf{lbl}^\star)$ that would have survived the private verification test of Game_3, we have $|\Pr[S_4] - \Pr[S_3]| \leq \Pr[F_4]$. Since $\Pr[S_4] = 0$ by construction, we are left with the task of bounding $\Pr[F_4]$.

Claim 2. The probability of event F_4 is at most $\Pr[F_4] \leq 1/(p-q)$, where q denotes the number of private verification queries.

Proof. The proof of the claim is a standard argument based on the smoothness of the projective hash function. If we consider the information that $\mathcal{A}$ can infer about the private evaluation key $\tau_v = \big(\boldsymbol{d} = (d_1, \ldots, d_n), \boldsymbol{e} = (e_1, \ldots, e_n)\big)$ by observing the CRS and the proof $\pi^\dagger = (z^\dagger, r^\dagger, u^\dagger, \pi_0^\dagger)$, it amounts to the first $2t+1$ rows of the left-hand-side member of the following linear system:

$$\begin{pmatrix} \mathbf{Y}^\top \\ \mathbf{W}^\top \\ \pi_0^\dagger \\ \pi_0^\star \end{pmatrix} = \begin{pmatrix} \mathbf{A} & \\ & \mathbf{A} \\ \log(\boldsymbol{v}^\dagger) & \alpha^\dagger \cdot \log(\boldsymbol{v}^\dagger) \\ \log(\boldsymbol{v}^\star) & \alpha^\star \cdot \log(\boldsymbol{v}^\star) \end{pmatrix} \cdot \begin{pmatrix} \mathbf{e}^\top \\ \mathbf{d}^\top \end{pmatrix} \tag{13}$$

Let us assume that $\boldsymbol{v}^\dagger, \boldsymbol{v}^\star \notin \mathcal{L}_\rho$. Since the above $(2t+2) \times 2n$ matrix has full rank when $\alpha^\star \neq \alpha$ (which is the case unless the failure event F_1 of Game_1 occurs), we see that the only value of $\pi_0^\star$ that would trick the private verification oracle $\mathsf{W}(\psi, \tau, ., .)$ of Game_3 into accepting $\pi^\star$ is completely independent of the information provided by the CRS and the simulated proof $\pi^\dagger$ for $\boldsymbol{v}^\dagger$. However, $\mathcal{A}$ can also take advantage of its private verification queries throughout the game. Without any verification query, $(\boldsymbol{e}, \boldsymbol{d})$ is constrained by the first $2t+1$ rows of (13) to live in a subspace of dimension $2(n-t)-1 \geq 1$ in $\mathbb{Z}_p^{2n}$, so that $\pi_0^\star$ has p equally likely values in $\mathcal{A}$'s view. Each private verification query provides $\mathcal{A}$ with an inequality, which allows it to rule out one candidate for the value of $\pi_0^\star$ that $\mathsf{W}(\psi, \tau_v, ., .)$ would accept. After q queries, $\mathcal{A}$ is thus left with $p-q$ equally likely candidates for $\pi_0^\star$. We thus find $\Pr[F_4] \leq 1/(p-q)$, as claimed. ■

Putting the above altogether, $\mathcal{A}$'s advantage is breaking the quasi-adaptive relative simulation-soundness property is bounded as

$$\mathbf{Adv}^{\mathrm{rss}}(\mathcal{A}) \leq \mathbf{Adv}^{\mathrm{CR}}(\lambda) + \mathbf{Adv}^{\mathrm{SDP}}(\lambda) + \frac{1}{2^\lambda - q}.$$

□

H Proof of KH-CCA Security for the Keyed-Homomorphic Scheme

Instead of relying on the simulation-soundness of the proof system in a modular manner, our proof of KH-CCA security uses a direct security analysis in order to obtain a tighter reduction.

Theorem 3. *The threshold keyed-homomorphic cryptosystem of Section 5 provides KH-CCA security in the sense of Definition 1 assuming that: (i) Σ is a strongly unforgeable one-time signature; (ii) The DLIN assumption holds in $\mathbb{G}$. Concretely, the advantage of any PPT adversary $\mathcal{A}$ is at most*

$$\mathbf{Adv}^{\text{kh-cca}}(\mathcal{A}) < (q_e+1)\cdot\mathbf{Adv}^{\text{suf-ots}}(\lambda) + \frac{3}{2}\cdot\mathbf{Adv}^{\text{DLIN}}(\lambda) + 7\cdot\sqrt{(q_e+1)\cdot(L+1)\cdot\mathbf{Adv}^{\text{DLIN}}(\lambda)\cdot\big(1-\frac{1}{p}\big)^{-1}} + \frac{1}{p},$$

where $\mathbf{Adv}^{\text{suf-ots}}(\lambda)$ denotes $\mathcal{A}$'s probability to break the strong unforgeability of Σ and q_e is the maximal number of evaluation queries involving derivatives of the challenge ciphertext.

Proof. The proof uses of a sequence of games starting with the real attack game and ending with a game where the adversary $\mathcal{A}$ has no advantage. For each i, we also denote by S_i the event that the challenger outputs 1 in Game_i.

Game_1**:** is the actual attack game with the only difference that the challenger $\mathcal{B}$ does not erase sk_{ot} during the key generation phase. In details, the adversary is given the public key PK and the set of verification keys $\mathbf{VK} = (VK_1, \ldots, VK_N)$. If $\mathcal{A}$ decides to query the RevHK oracle at some point, $\mathcal{B}$ reveals SK_h. At each corruption query $i \in \{1, \ldots, N\}$, $\mathcal{B}$ reveals the queried private key share $SK_i = (P_1(i), P_2(i), P(i))$. At each decryption query, $\mathcal{B}$ faithfully runs the real shared decryption algorithm. At each evaluation query, $\mathcal{A}$ supplies two ciphertexts $C^{(1)}, C^{(2)}$ which are processed by $\mathcal{B}$ as in the evaluation algorithm. We denote by $C_1^\dagger, \ldots, C_{q_e}^\dagger$ the outputs of the $\mathsf{Eval}(SK_h, .)$ oracle when the latter is queried on a pair $(C^{(1)}, C^{(2)})$ such that $C^{(j)} \in \mathcal{D}$ for some $j \in \{1, 2\}$ (in other words, $\{C_l^\dagger\}_{l=1}^{q_e}$ are the results of evaluation queries which increase $|\mathcal{D}|$). We also denote by $\mathsf{SVK}_1^\dagger, \ldots, \mathsf{SVK}_{q_e}^\dagger$ the one-time verification keys appearing in these ciphertexts and assume w.l.o.g. that they are chosen by $\mathcal{B}$ at the very beginning of the game.

When the first phase is over, the adversary $\mathcal{A}$ chooses two distinct messages $M_0, M_1 \in \mathbb{G}$ and obtains $C^\star = (\mathsf{SVK}^\star, C_0^\star, C_1^\star, C_2^\star, C_3^\star, Z^\star, R^\star, U^\star, \boldsymbol{C}_z^\star, \boldsymbol{C}_r^\star, \boldsymbol{C}_u^\star, \boldsymbol{\pi}_1^\star, \boldsymbol{\pi}_2^\star, \sigma^\star)$ which is an encryption of M_β, for some random coin $\beta \xleftarrow{R} \{0, 1\}$ flipped by $\mathcal{B}$.

In the second phase, $\mathcal{A}$ makes more decryption, evaluation and corruption queries under the restriction of not asking for a partial decryption of a ciphertext in $\mathcal{D}$ or for more than $t-1$ private key shares throughout the entire game. Eventually, $\mathcal{A}$ halts and outputs $\beta' \in \{0, 1\}$. The challenger $\mathcal{B}$ outputs 1 if and only if $\beta = \beta'$. We denote this event by S_1.

Game_2**:** This game is identical to Game_1 with the difference that the challenger $\mathcal{B}$ rejects all decryption queries involving ciphertexts $C = (\mathsf{SVK}, C_0, C_1, C_2, C_3, Z, R, U, \boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \sigma)$ such that $\mathsf{SVK} \in \{\mathsf{SVK}^\star, \mathsf{SVK}_1^\dagger, \ldots, \mathsf{SVK}_{q_e}^\dagger\}$. It also returns $\perp$ at each evaluation query $(C^{(1)}, C^{(2)})$ for which there exists $j \in \{1, 2\}$ for which $C^{(j)}$ contains a verification key $\mathsf{SVK}^{(j)}$ such that $\mathsf{SVK}^{(j)} = \mathsf{SVK}_l^\dagger$, for some $\mathsf{SVK}_l^\dagger \in \{\mathsf{SVK}^\star, \mathsf{SVK}_1^\dagger, \ldots, \mathsf{SVK}_{q_e}^\dagger\}$, but $C^{(j)} \neq C_l^\dagger$.

If we define F_2 to be the event that $\mathcal{B}$ rejects a ciphertext that would not have been rejected in Game_1, we see that Game_2 and Game_1 proceed identically until event F_2 occurs. We thus have the inequality $|\Pr[S_2] - \Pr[S_1]| \leq \Pr[F_2]$. Moreover, event F_2 would imply a breach in the strong unforgeability of the one-time signature. Indeed, since $\mathcal{A}$ is not allowed to query the partial decryption of any ciphertext in $\mathcal{D}$, a standard argument allows proving the inequality $|\Pr[S_2] - \Pr[S_1]| \leq \Pr[F_2] \leq (q_e+1)\cdot\mathbf{Adv}^{\text{suf-ots}}(\lambda)$. In the forthcoming games, we will assume that $\mathsf{SVK} \notin \{\mathsf{SVK}^\star, \mathsf{SVK}_1^\dagger, \ldots, \mathsf{SVK}_{q_e}^\dagger\}$ at each decryption query.

Game_3**:** We modify the generation of the challenge ciphertext $C^\star$. Namely, instead of computing $C_0^\star = M_\beta \cdot X_1^\theta \cdot X_2^\theta$, using the encryption exponents $\theta_1 = \log_f(C_1^\star)$ and $\theta_2 = \log_h(C_2^\star)$, $\mathcal{B}$ uses the private key (x_0, x_1, x_2) and computes $C_0^\star = M_\beta \cdot C_1^{\star x_1} \cdot C_2^{\star x_2} \cdot C_3^{\star x_0}$. Likewise, instead of using

(θ_1, θ_2) to derive the triple $(z^\star, r^\star, u^\star)$ at step 3 of the encryption algorithm, $\mathcal{B}$ uses the simulation trapdoor $\mathsf{sk}_{rand} = \{(\chi_i, \gamma_i, \delta_i)\}_{i=1}^3$ of the USS proof system and computes

$$z^\star = \prod_{i=1}^{3} C_i^{\star -\chi_i} \qquad r^\star = \prod_{i=1}^{3} C_i^{\star -\gamma_i} \qquad u^\star = \prod_{i=1}^{3} C_i^{\star -\delta_i}. \tag{14}$$

Finally, $(Z^\star, R^\star, U^\star)$ is generated using $\mathsf{sk}_{ot} = \{(\varphi_i, \vartheta_i, \varpi_i)\}_{i=1}^3$ as

$$Z^\star = \prod_{i=1}^{3} C_i^{\star -\varphi_i} \qquad R^\star = \prod_{i=1}^{3} C_i^{\star -\vartheta_i} \qquad U^\star = \prod_{i=1}^{3} C_i^{\star -\varpi_i}.$$

Then, $\mathcal{B}$ conducts steps 4-6 as in the actual encryption algorithm. It is easy to see that this change does not modify $\mathcal{A}$'s view since $C_0^\star$ still equals $C_0^\star = M_\beta \cdot X_1^{\theta_1} \cdot X_2^{\theta_2}$ and the distributions of $(z^\star, r^\star, u^\star)$ and $(Z^\star, R^\star, U^\star)$ remain the same as in Game_2. We thus have $\Pr[S_3] = \Pr[S_2]$.

Game_4: This game is identical to Game_3 with a new modification in the challenge ciphertext. Namely, instead of setting $C_3^\star = g^{\theta_1+\theta_2}$, where $\theta_1 = \log_f(C_1^\star)$ and $\theta_2 = \log_h(C_2^\star)$, we choose it as $C_3^\star \stackrel{R}{\leftarrow} \mathbb{G}$. At the third step of the encryption algorithm, the linearly homomorphic signature $(z^\star, r^\star, u^\star)$ is computed according to (14), as previously. Under the DLIN assumption in $\mathbb{G}$, this modification should not significantly alter $\mathcal{A}$'s behavior. In particular, we have $|\Pr[S_4] - \Pr[S_3]| \leq \mathbf{Adv}^{\mathrm{DLIN}}(\lambda)$.

Game_5: From this point forward, we make explicit use of the discrete logarithms $\alpha_f = \log_g(f)$ and $\alpha_h = \log_g(h)$, which is allowed since we are done with the transition consisting in tampering with $C_3^\star$ in the challenge ciphertext. In Game_5, we first change the treatment of ciphertexts $C = (\mathsf{SVK}, C_0, C_1, C_2, C_3, Z, R, U, \boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \sigma)$ involved in *pre-challenge* decryption and evaluation queries. Namely, $\mathcal{B}$ simply ignores the linearly homomorphic signatures contained these ciphertexts and returns $\perp$ if $C_3 \neq C_1^{1/\alpha_f} \cdot C_2^{1/\alpha_h}$. Otherwise, it responds as in earlier games.

If we call F_5 the event that $\mathcal{B}$ rejects a ciphertext that would not have been rejected in Game_4, Game_5 and Game_4 are clearly identical until F_5 occurs, so that $|\Pr[S_5] - \Pr[S_4]| \leq \Pr[F_5]$. At the same time, Lemma 1 shows that $\Pr[F_5] \leq \frac{1}{2} \cdot \mathbf{Adv}^{\mathrm{DLIN}}(\lambda) + 1/p$.

The proof of Lemma 1 implies that, even if $\mathcal{A}$ chooses to expose the evaluation key SK_h at the very beginning of the game, it should not be able to create valid-looking ill-formed ciphertexts before the challenge phase unless the DLIN assumption is false. This will help us prove that the scheme remains IND-CCA1 if SK_h is revealed to the adversary when the game begins.

Game_6: We now modify the treatment of *post-challenge* queries and introduce yet another event F_6. In this game, the challenger $\mathcal{B}$ halts and outputs a random bit in the event that the adversary $\mathcal{A}$ manages to query the partial decryption oracle or the evaluation oracle on a ciphertext $C^\diamond = (\mathsf{SVK}^\diamond, C_0^\diamond, C_1^\diamond, C_2^\diamond, C_3^\diamond, Z^\diamond, R^\diamond, U^\diamond, \boldsymbol{C}_z^\diamond, \boldsymbol{C}_r^\diamond, \boldsymbol{C}_u^\diamond, \boldsymbol{\pi}_1^\diamond, \boldsymbol{\pi}_2^\diamond, \sigma^\diamond)$ where $\mathsf{SVK}^\diamond \notin \{\mathsf{SVK}^\star, \mathsf{SVK}_1^\dagger, \ldots, \mathsf{SVK}_{q_e}^\dagger\}$ and $C_3^\diamond \neq C_1^{\diamond 1/\alpha_f} \cdot C_2^{\diamond 1/\alpha_h}$ although $(\boldsymbol{C}_z^\diamond, \boldsymbol{C}_r^\diamond, \boldsymbol{C}_u^\diamond, \boldsymbol{\pi}_1^\diamond, \boldsymbol{\pi}_2^\diamond)$ is a valid linearly homomorphic signature on the vector $(C_1^\diamond, C_2^\diamond, C_3^\diamond)$. We say that $C^\diamond$ is a *fatal* query in this case. Since Game_6 is identical to Game_5 until event F_6 occurs, we have $|\Pr[S_6] - \Pr[S_5]| \leq \Pr[F_6]$. Lemma 2 demonstrates that the DLIN assumption can be broken in the group $\mathbb{G}$ if event F_6 occurs with non-negligible probability. More precisely, Lemma 2 shows that $\Pr[F_6] \leq 7 \cdot \sqrt{(q_e+1) \cdot (L+1) \cdot \mathbf{Adv}^{\mathrm{DLIN}}(\lambda) \cdot \left(1 - \frac{1}{p}\right)^{-1}}$.

Game_7: We now modify the partial decryption oracle and replace the non-interactive proofs contained in decryption shares $\hat{\mu}_i$ by simulated NIZK proofs. This entails to turn $(\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{f}_3)$ into a perfectly hiding Groth-Sahai CRS (where $\boldsymbol{f}_3$ is in the span of $\boldsymbol{f}_1$ and $\boldsymbol{f}_2$) and non-interactive proofs for multi-exponentiation equations are simulated using the trapdoor of the simulated CRS. Under the DLIN assumption, this change is not noticeable to $\mathcal{A}$ and we have $|\Pr[S_7] - \Pr[S_6]| \leq \mathbf{Adv}^{\mathrm{DLIN}}(\lambda)$.

In Game_7, it is easy to see that $\mathcal{A}$ has no advantage whatsoever, so that $\Pr[S_7] = 1/2$. Indeed, in the challenge phase, we have $(C_1^\star, C_2^\star, C_3^\star) = (f^{\theta_1}, h^{\theta_2}, g^{\theta_1+\theta_2+\theta_3})$, where $\theta_1, \theta_2, \theta_3 \in_R \mathbb{Z}_p$. This implies

that $C_0^\star$ can be written as $C_0^\star = M_\beta \cdot X_1^{\theta_1} \cdot X_2^{\theta_2} \cdot g^{\theta_3 \cdot x_0}$. The latter equality implies that, as long as $x_0 \in \mathbb{Z}_p$ remains independent of $\mathcal{A}$'s view, so does the challenger's bit $\beta \in \{0,1\}$.

To see why $\mathcal{A}$ does not learn anything about $x_0 \in \mathbb{Z}_p$, we first note that the homomorphic evaluation key SK_h is independent of x_0, so that homomorphic evaluation queries leak nothing about it. We also remark that, in Game_7, decryption shares $\hat{\mu}_i$ contain NIZK proofs that are simulated without using private key shares. Hence, as long as $\mathcal{A}$ does not corrupt more than $t-1$ servers, the only possible way to infer information about $x_0 = P(0)$ is to trick the partial decryption oracle into accepting an invalid ciphertext. However, in Game_7, all invalid ciphertexts are explicitly rejected.

We thus find the announced result

$$|\Pr[S_1] - \frac{1}{2}| < (q_e+1) \cdot \mathbf{Adv}^{\text{suf-ots}}(\lambda) + \frac{3}{2} \cdot \mathbf{Adv}^{\text{DLIN}}(\lambda) + 7 \cdot \sqrt{(q_e+1) \cdot (L+1) \cdot \mathbf{Adv}^{\text{DLIN}}(\lambda) \cdot \big(1 - \frac{1}{p}\big)^{-1}} + \frac{1}{p}.$$

□

Lemma 1. *In* Game_5*, the probability of event* F_5 *is at most* $\Pr[F_5] \leq \frac{1}{2} \cdot \mathbf{Adv}^{\text{DLIN}}(\lambda) + \frac{1}{p}$.

Proof. We show that, if event F_5 occurs with non-negligible probability in Game_5, there exists an efficient algorithm $\mathcal{B}$ that solves a SDP instance (G_z, G_r, H_z, H_u) with nearly the same probability. In turn, $\mathcal{B}$ implies a distinguisher for the DLIN assumption in $\mathbb{G}$.

Algorithm $\mathcal{B}$ generates public key components are defined as in the actual scheme. In particular, $\mathcal{B}$ sets $G_i = {G_z}^{\varphi_i} {G_r}^{\vartheta_i}$ and $H_i = {H_z}^{\varphi_i} {H_u}^{\varpi_i}$ with $\varphi_i, \vartheta_i, \varpi_i \xleftarrow{R} \mathbb{Z}_p$ for $i \in \{1,2,3\}$.

Throughout the game, the reduction $\mathcal{B}$ answers $\mathcal{A}$'s decryption queries in the same way as in Game_5. Moreover, since $\mathcal{B}$ has generated $(\mathsf{sk}_{rand}, \mathsf{pk}_{rand})$ faithfully, it is able to consistently reveal the evaluation key SK_h in case $\mathcal{A}$ decides to corrupt the evaluator. If event F_5 occurs with non-negligible probability, we know that, before the challenge phase, $\mathcal{A}$ must query the partial decryption or the homomorphic evaluation of a ciphertext $C = (\mathsf{SVK}, C_0, C_1, C_2, C_3, Z, R, U, \boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \sigma)$ such that (Z, R, U) is a valid one-time linearly homomorphic signature on (C_1, C_2, C_3) although (C_1, C_2, C_3) is outside the span of $(f, 1, g)$ and $(1, h, g)$. When algorithm $\mathcal{B}$ detects this event (by observing that $C_3 \neq C_1^{1/\alpha_f} C_2^{1/\alpha_h}$), it computes its own signature

$$(Z^\dagger, R^\dagger, U^\dagger) = (\prod_{i=1}^{3} {C_i}^{-\varphi_i}, \prod_{i=1}^{3} {C_i}^{-\vartheta_i}, \prod_{i=1}^{3} {C_i}^{-\varpi_i}) \tag{15}$$

on (C_1, C_2, C_3). We claim that, with overwhelming probability,

$$(Z^\ddagger, R^\ddagger, U^\ddagger) = \Big(\frac{Z}{Z^\dagger}, \frac{R}{R^\dagger}, \frac{U}{U^\dagger}\Big)$$

is a non-trivial solution to the SDP instance since $Z^\ddagger \neq 1_{\mathbb{G}}$ with overwhelming probability.

Indeed, we remark that the vector $(\varphi_1, \varphi_2, \varphi_3)$ is independent of $\mathcal{A}$'s view before the challenge phase. Consequently, since (C_1, C_2, C_3) is linearly independent of $(f, 1, g)$ and $(1, h, g)$, the adversary $\mathcal{A}$ is only able to predict the value $Z^\dagger$ of (15) with probability $1/p$. Given that we also have the inequality $\mathbf{Adv}^{\text{SDP}}(\lambda) \leq \frac{1}{2} \cdot \mathbf{Adv}^{\text{DLIN}}(\lambda)$, we thus find $\Pr[F_5] \leq \frac{1}{2} \cdot \mathbf{Adv}^{\text{DLIN}}(\lambda) + \frac{1}{p}$ as claimed. □

In the transition from Game_5 to Game_6, we could rely on the unbounded simulation-soundness of the underlying QA-NIZK proof to argue that fatal decryption or evaluation queries only occur with negligible probability. To do this, we would have to build a reduction algorithm that interacts with a simulation-soundness challenger for a given matrix $\rho \in \mathbb{G}^{2\times 3}$ and simultaneously emulates $\mathcal{A}$'s challenger in the KH-CCA game. Since the reduction would not have the matrix $\mathbf{A} \in \mathbb{Z}_p^{2\times 3}$, it would have no way to detect fatal queries. Consequently, the reduction would have to guess this query, which would introduce an extra degradation factor in the reduction.

Lemma 2. *In* Game_6*, the probability of event* F_6 *is at most*

$$\Pr[F_6] \leq 7 \cdot \sqrt{(q_e+1)\cdot(L+1)\cdot \mathbf{Adv}^{\mathrm{DLIN}}(\lambda)\cdot\big(1-\frac{1}{p}\big)^{-1}}.$$

Proof. The key argument of the proof is that, conditionally on a certain desirable event, the evaluation oracle $\mathsf{Eval}(SK_h,.)$ will never information-theoretically reveal its evaluation key SK_h.

Assuming that event F_6 occurs with non-negligible probability in Game_6, we show that there exists a distinguisher for the DLIN assumption in $\mathbb{G}$. To this end, we consider a subsequence of games starting with Game_6 and ending with $\mathsf{Game}_{6.2}$. For each $j \in \{0,1,2\}$, we define $F_{6.j}$ as the counterpart of event F_6 in $\mathsf{Game}_{6.j}$ (note that $F_{6.j}$ is efficiently detectable for each j). We first show that, as long as the DLIN assumption holds, if $\Pr[F_6]$ is non-negligible, so is $\Pr[F_{6.2}]$.

$\mathsf{Game}_{6.0}$: This game is identical to Game_6 but we modify the generation of pk_{rand} when the public key is set up. Namely, the vectors $(\boldsymbol{f}_1, \boldsymbol{f}_2, \{\boldsymbol{f}_{3,i}\}_{i=0}^L)$ are chosen by setting $\boldsymbol{f}_1 = (f_1, 1_{\mathbb{G}}, g)$ and $\boldsymbol{f}_2 = (1_{\mathbb{G}}, f_2, g)$ where $f_1, f_2 \xleftarrow{R} \mathbb{G}$ are chosen at random. As for $\{\boldsymbol{f}_{3,i}\}_{i=0}^L$, they are obtained as

$$\begin{aligned} \boldsymbol{f}_{3,0} &= \boldsymbol{f}_1^{\xi_{0,1}} \cdot \boldsymbol{f}_2^{\xi_{0,2}} \cdot (1,1,g)^{\xi_{0,3}} \cdot (1,1,g)^{\mu\cdot\zeta-\rho_0} \\ \boldsymbol{f}_{3,i} &= \boldsymbol{f}_1^{\xi_{i,1}} \cdot \boldsymbol{f}_2^{\xi_{i,2}} \cdot (1,1,g)^{\xi_{i,3}} \cdot (1,1,g)^{-\rho_i}, \qquad\qquad i \in \{1,\ldots,L\} \end{aligned} \tag{16}$$

with $\mu \xleftarrow{R} \{0,\ldots,L\}$, $\xi_{0,1}, \xi_{1,1}, \ldots, \xi_{L,1} \xleftarrow{R} \mathbb{Z}_p$, $\xi_{0,2}, \xi_{1,2}, \ldots, \xi_{L,2} \xleftarrow{R} \mathbb{Z}_p$, $\xi_{0,3}, \xi_{1,3}, \ldots, \xi_{L,3} \xleftarrow{R} \mathbb{Z}_p$ and $\rho_0, \rho_1, \ldots, \rho_L \xleftarrow{R} \{0,\ldots,\zeta-1\}$, with $\zeta = 2(q_e+1)$ and where q_e is the number of evaluation queries that increase the cardinality of $\mathcal{D}$. Note that this change is only conceptual since the distribution of $\{\boldsymbol{f}_{3,i}\}_{i=0}^L$ has not changed since Game_6. We thus have $\Pr[F_{6.0}] = \Pr[F_6]$.

$\mathsf{Game}_{6.1}$**:** This game is like $\mathsf{Game}_{6.0}$ but we consider another event Good which causes the challenger $\mathcal{B}$ to abort and output a random bit if it does *not* occur. Let $\mathsf{SVK}_1^\dagger, \ldots, \mathsf{SVK}_{q_e}^\dagger$ be the distinct one-time verification keys appearing in outputs of the $\mathsf{Eval}(SK_h,.)$ oracle when the latter is invoked on a ciphertext in $\mathcal{D}$. Let also $\mathsf{SVK}^\star$ be the verification key involved in the challenge ciphertext $C^\star$ and let $\mathsf{SVK}^\diamond$ be the one involved in the first fatal query $C^\diamond$. We know that $\mathsf{SVK}^\diamond \notin \{\mathsf{SVK}^\star, \mathsf{SVK}_1^\dagger, \ldots, \mathsf{SVK}_{q_e}^\dagger\}$. For each verification key $\mathsf{SVK} \in \{0,1\}^L$, we consider the function $J(\mathsf{SVK}) = \mu\cdot\zeta - \rho_0 - \sum_{i=1}^L \rho_i \cdot \mathsf{SVK}[i]$. We also define Good to be the event that

$$J(\mathsf{SVK}^\diamond) = 0 \qquad \wedge \qquad J(\mathsf{SVK}^\star) \neq 0 \qquad \wedge \qquad \bigwedge_{j\in\{1,\ldots,q_e\}} J(\mathsf{SVK}_j^\dagger) \neq 0. \tag{17}$$

We remark that the random exponents $\rho_0, \rho_1, \ldots, \rho_L$ are chosen independently of $\mathcal{A}$'s view: this means that the simulator could equivalently define $\{\boldsymbol{f}_{3,i}\}_{i=0}^L$ first and only choose $\{\rho_i\}_{i=0}^L$ – together with values $\{\xi_{3,i}\}_{i=0}^L$ explaining the $\{\boldsymbol{f}_{3,i}\}_{i=0}^L$ – at the very end of the game, when $\mathsf{SVK}^\star, \mathsf{SVK}_1^\dagger, \ldots, \mathsf{SVK}_{q_e}^\dagger, \mathsf{SVK}^\diamond$ have been defined. The same analysis as [54] (using the simplifications of Bellare and Ristenpart [6]) shows that $\Pr[F_{6.1} \wedge \mathsf{Good}] \geq \Pr[F_{6.0}]^2/(27\cdot(q_e+1)\cdot(L+1))$.

This follows from the fact that, for any set of queries, a lower bound on the probability of event Good is $1/(2(q_e+1)(L+1))$. Indeed, from the known results [54, 35] on the programmability of Waters' hash function, we know that the probability, taken over the choice of $(\mu, \rho_0, \ldots, \rho_L)$, to meet the conditions (17) is at least $1/(2(q_e+1)(L+1))$.

$\mathsf{Game}_{6.2}$**:** We modify again the way to compute pk_{rand} in the generation of the public key. Namely, the vectors $\boldsymbol{f_1} = (f_1, 1_{\mathbb{G}}, g)$, $\boldsymbol{f_2} = (1_{\mathbb{G}}, f_2, g)$ are chosen as before. However, instead of generating $\{\boldsymbol{f}_{3,i}\}_{i=0}^L$ as in $\mathsf{Game}_{6.1}$, we set them as

$$\begin{aligned} \boldsymbol{f}_{3,0} &= \boldsymbol{f}_1^{\xi_{0,1}} \cdot \boldsymbol{f}_2^{\xi_{0,2}} \cdot (1,1,g)^{\mu\cdot\zeta-\rho_0} \\ \boldsymbol{f}_{3,i} &= \boldsymbol{f}_1^{\xi_{i,1}} \cdot \boldsymbol{f}_2^{\xi_{i,2}} \cdot (1,1,g)^{-\rho_i}, \qquad\qquad i \in \{1,\ldots,L\} \end{aligned} \tag{18}$$

which amounts to setting $\xi_{0,3} = \xi_{1,3} = \ldots = \xi_{L,3} = 0$. Clearly, $\{\boldsymbol{f}_{3,i}\}_{i=0}^{L}$ are no longer uniform in the span of $\big(\boldsymbol{f}_1, \boldsymbol{f}_2, (1,1,g)\big)$. Still, this change should have no noticeable effect on $\mathcal{A}$ if the DLIN assumption holds in $\mathbb{G}$. Concretely, if a fatal decryption/evaluation query occurs with substantially different probabilities in $\mathsf{Game}_{6.2}$ and $\mathsf{Game}_{6.1}$, we can construct a DLIN distinguisher $\mathcal{B}^{\mathrm{DLIN}}$ in the group $\mathbb{G}$ (recall that the reduction can detect fatal queries using $\alpha_f = \log_g(f)$ and $\alpha_h = \log_g(h)$). This distinguisher uses the random self-reducibility of DLIN to construct many independent-looking instances from the same distribution out of a given instance. For this reason, we can write $|\Pr[F_{6.2} \wedge \mathsf{Good}] - \Pr[F_{6.1} \wedge \mathsf{Good}]| \leq \mathbf{Adv}_{\mathcal{B}^{\mathrm{DLIN}}}(\lambda)$.

In $\mathsf{Game}_{6.2}$, we show that an occurrence of event $F_{6.2}$ implies an algorithm $\mathcal{B}$ solving a given SDP instance (g_z, g_r, h_z, h_u) with non-negligible probability, which *a fortiori* breaks the DLIN assumption in $\mathbb{G}$ as the latter is implied by SDP.

By hypothesis, we know that the adversary $\mathcal{A}$ somehow manages to produce a fatal decryption/evaluation query on a ciphertext $C^\diamond$ for which $(C_1^\diamond, C_2^\diamond, C_3^\diamond)$ is outside the span of $(f, 1_\mathbb{G}, g)$ and $(1_\mathbb{G}, h, g)$ but $(\boldsymbol{C}_z^\diamond, \boldsymbol{C}_r^\diamond, \boldsymbol{C}_u^\diamond, \boldsymbol{\pi}_1^\diamond, \boldsymbol{\pi}_2^\diamond) \in \mathbb{G}^{15}$ satisfies the verification equations. At this point, if the event Good introduced in $\mathsf{Game}_{6.1}$ occurs, we must have $J(\mathsf{SVK}^\diamond) = 0$, which implies that $\boldsymbol{f}_{\mathsf{SVK}^\diamond} = \boldsymbol{f}_{3,0} \cdot \prod_{i=1}^{L+1} \boldsymbol{f}_{3,i}^{\mathsf{SVK}^\diamond[i]}$ lies in $\mathrm{span}(\boldsymbol{f}_1, \boldsymbol{f}_2)$. Consequently, $\boldsymbol{C}_z^\diamond$, $\boldsymbol{C}_r^\diamond$ and $\boldsymbol{C}_u^\diamond$ are necessarily perfectly binding and extractable commitments. Using $(\log_g(f_1), \log_g(f_2))$, $\mathcal{B}$ can thus extract the committed group elements $(z^\diamond, r^\diamond, u^\diamond) \in \mathbb{G}^3$ by BBS-decrypting the ciphertexts $(\boldsymbol{C}_z^\diamond, \boldsymbol{C}_r^\diamond, \boldsymbol{C}_u^\diamond)$. Since $(\boldsymbol{\pi}_1^\diamond, \boldsymbol{\pi}_2^\diamond)$ are perfectly sound Groth-Sahai proofs, the extracted elements $(z^\diamond, r^\diamond, u^\diamond)$ necessarily satisfy

$$1_{\mathbb{G}_T} = e(g_z, z^\diamond) \cdot e(g_r, r^\diamond) \cdot \prod_{i=1}^{3} e(g_i, C_i^\diamond) = e(h_z, z^\diamond) \cdot e(h_u, u^\diamond) \cdot \prod_{i=1}^{3} e(h_i, C_i^\diamond). \tag{19}$$

Having extracted $(z^\diamond, r^\diamond, u^\diamond)$, $\mathcal{B}$ also computes

$$z^\dagger = \prod_{i=1}^{3} C_i^{\diamond - \chi_i} \qquad r^\dagger = \prod_{i=1}^{3} C_i^{\diamond - \gamma_i} \qquad u^\dagger = \prod_{i=1}^{3} C_i^{\diamond - \delta_i}, \tag{20}$$

so that $(z^\dagger, r^\dagger, u^\dagger)$ also satisfies (19). Since $(z^\dagger, r^\dagger, u^\dagger)$ and $(z^\diamond, r^\diamond, u^\diamond)$ both satisfy (19), the triple

$$(z^\ddagger, r^\ddagger, u^\ddagger) = \Big(\frac{z^\diamond}{z^\dagger}, \frac{r^\diamond}{r^\dagger}, \frac{u^\diamond}{u^\dagger}\Big)$$

satisfies $e(g_z, z^\ddagger) \cdot e(g_r, r^\ddagger) = e(h_z, z^\ddagger) \cdot e(h_u, u^\ddagger) = 1_{\mathbb{G}_T}$. To conclude the proof, we argue that $z^\ddagger \neq 1_\mathbb{G}$ with overwhelming probability.

To do this, we observe that, if the event Good defined in $\mathsf{Game}_{6.1}$ actually comes about, then $\mathcal{B}$ never leaks any more information about (χ_1, χ_2, χ_3) than $\mathcal{A}$ can infer by just observing $\{(z_j, r_j, u_j)\}_{j=1}^2$ in the public key. Indeed, in this case we have $J(\mathsf{SVK}^\star) \neq 0$ and $J(\mathsf{SVK}_j^\dagger) \neq 0$ for each $j \in \{1, \ldots, q_e\}$. This means that, in the challenge ciphertext and all its homomorphic evaluations, the proofs $(\boldsymbol{\pi_1}, \boldsymbol{\pi_2})$ are perfectly WI as they are generated for a perfectly hiding Groth-Sahai CRS. For these ciphertexts, the built-in homomorphic signatures $(\boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi_2})$ leak nothing about the specific vector $(\chi_1, \chi_2, \chi_3) \in \mathbb{Z}_p^3$ used by $\mathcal{B}$. As a consequence, we can apply the same arguments as in the proof of Lemma 1 when it comes to argue that $z^\dagger \neq z^\diamond$ with probability $1 - 1/p$. We thus find

$$\Pr[F_{6.2} \wedge \mathsf{Good}] = \mathbf{Adv}_{\mathcal{B}}^{\mathrm{SDP}}(\lambda) \cdot \Big(1 - \frac{1}{p}\Big)^{-1}.$$

In turn, $\mathcal{B}$ implies a PPT distinguisher $\mathcal{B}^{\mathrm{DLIN}'}$ for the DLIN assumption such that we have the inequality $\Pr[F_{6.2} \wedge \mathsf{Good}] < \frac{1}{2} \cdot \mathbf{Adv}_{\mathcal{B}^{\mathrm{DLIN}'}}(\lambda) \cdot \big(1 - \frac{1}{p}\big)^{-1}$. If $\mathbf{Adv}^{\mathrm{DLIN}}(\lambda)$ denotes the maximal advantage of any PPT distinguisher against the DLIN assumption in $\mathbb{G}$, the probability to have $F_{6.1} \wedge \mathsf{Good}$ can

be bounded as $\Pr[F_{6.1} \wedge \mathsf{Good}] \leq \frac{3}{2} \cdot \mathbf{Adv}^{\mathrm{DLIN}}(\lambda) \cdot \left(1 - \frac{1}{p}\right)^{-1}$ in $\mathsf{Game}_{6.1}$. This eventually yields the stated result

$$\Pr[F_6] \leq 7 \cdot \sqrt{(q_e + 1) \cdot (L + 1) \cdot \mathbf{Adv}^{\mathrm{DLIN}}(\lambda) \cdot \left(1 - \frac{1}{p}\right)^{-1}}.$$

□

I More Efficient Adaptively Secure CCA2-Secure Threshold Cryptosystems from the DLIN and *k*-Linear Assumptions

As a use case for our relatively sound QA-NIZK proofs, we can construct a new robust non-interactive threshold encryption scheme based on the DLIN assumption and prove it secure against chosen-ciphertext attacks in the adaptive corruption setting [16, 28].

Threshold cryptosystems were initially suggested in [13, 22, 23]. In the static corruption setting, several non-interactive CCA-secure threshold systems have been described in the random oracle model [53, 27] and in the standard model [9, 14, 55].

Adaptively secure distributed cryptosystems with chosen-ciphertext security were proposed in [36, 1] but they require some interaction during the decryption process. Non-interactive solutions were put forth in [41, 42] but, as we will see, they are less efficient than the solution proposed here.

Consider the DLIN-based cryptosystem based on 1-universal hash proof systems where the ciphertext $(C_1, C_2, C_3, C_0) = (f^{\theta_1}, h^{\theta_2}, g^{\theta_1+\theta_2}, M \cdot X_1^{\theta_1} \cdot X_2^{\theta_2})$ is decrypted as $M = C_0 \cdot C_1^{-x_1} C_2^{-x_2} C_3^{-x_0}$, where $(X_1, X_2) = (f^{x_1} g^{y}, h^{x_2} g^{x_0})$ is the public key and (x_1, x_2, x_0) is the private key. In [42], chosen-ciphertext security was achieved using a *publicly* verifiable one-time simulation-sound proof of well-formedness for (C_1, C_2, C_3). In the security proof, the one-time simulation-soundness property guarantees that the adversary is unable to trick the decryption oracle into returning the decryption of an invalid ciphertext, by generating a fake proof for an invalid triple (C_1, C_2, C_3). For this reason, the specific private key (x_1, x_2, x_0) used by the reduction remains perfectly hidden. Consequently, if the challenge ciphertext is computed by choosing a random tuple $(C_1, C_2, C_3) \in_R \mathbb{G}^3$ and computing $C_0 = M \cdot {C_1}^{x_1} \cdot {C_2}^{x_2} \cdot {C_3}^{x_0}$, the plaintext M is independent of the adversary's view. To prove adaptive security in the threshold setting, [42] took advantage of the fact that the private key (more precisely, all private key shares) is known to the reduction at all times in the Cramer-Shoup paradigm.

A similar approach was taken in[6] [37], where a different method was used to achieve a form of one-time simulation-soundness. In combination with relatively sound proofs [37], the techniques of Jutla and Roy [38] reduce the size of ciphertexts to 9 group elements under the DLIN assumption.

Here, as already suggested in [43], we obtain shorter ciphertexts by using linearly homomorphic signatures. We include in the public key the verification key of a one-time linearly homomorphic SPS for $n = 3$ as well as signatures on $(f, 1_{\mathbb{G}}, g)$ and $(1_{\mathbb{G}}, h, g)$. This allows publicly deriving a homomorphic signature (z, r, u) on the vector $(C_1, C_2, C_3) = (f^{\theta_1}, h^{\theta_2}, g^{\theta_1+\theta_2})$ and each ciphertext consists of $(z, r, u, C_0, C_1, C_2, C_3)$. In the security proof, the signature (z, r, u) serves as evidence that (C_1, C_2, C_3) has the right form at each pre-challenge decryption query: in order to generate a proof for a false statement, the adversary has to break the security of the homomorphic signature, by deriving a signature on a vector (C_1, C_2, C_3) outside the span of $(f, 1, g)$ and $(1, h, g)$.

While this technique does provide IND-CCA1 security, the scheme remains malleable and thus vulnerable to post-challenge decryption queries. This is where the relatively sound proof system of Section 4 comes into play. By using (C_0, C_1, C_2, C_3) as a label in the relatively sound proof that (C_1, C_2, C_3) lives in $\mathrm{span}((f, 1, g), (1, h, g))$, we can make sure that, with all but negligible probability, the reduction will never accept a proof for a malformed (C_1, C_2, C_3) after the challenge phase without breaking the DLIN assumption. The key idea of the techniques of [37] is to guarantee that

[6] Although it was not mentioned in [37], relatively sound proofs can be used to acquire CCA2 security in the threshold setting as well, as will be emphasized later on.

the adversary will not be able to send a decryption query for which the private verifier and the public verifier disagree on (C_1, C_2, C_3).

I.1 Construction

In the threshold setting, the construction can be seen as a DLIN-based version of the Cramer-Shoup encryption scheme [20] (which is identical to the scheme in [52]), where the ciphertext components (C_1, C_2, C_3) and the designated verifier proof C_4 are additionally signed using a homomorphic signature. The scheme goes as follows.

Keygen(λ, t, N)**:** choose bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$ and do the following.

1. Choose $g, f, h \xleftarrow{R} \mathbb{G}$.
2. Choose random $x_0, x_1, x_2 \xleftarrow{R} \mathbb{Z}_p$, $y_0, y_1, y_2 \xleftarrow{R} \mathbb{Z}_p$ and $w_0, w_1, w_2 \xleftarrow{R} \mathbb{Z}_p$ in order to compute $X_1 = f^{x_1} g^{x_0}$, $X_2 = h^{x_2} g^{x_0}$, $Y_1 = f^{y_1} g^{y_0}$, $Y_2 = h^{y_2} g^{y_0}$ and $W_1 = f^{w_1} g^{w_0}$, $W_2 = h^{w_2} g^{w_0}$.
3. Generate a Groth-Sahai CRS $(\boldsymbol{f_1}, \boldsymbol{f_2}, \boldsymbol{f_3})$ to be used for proving the validity of decryption shares. Namely, choose $f_1, f_2 \xleftarrow{R} \mathbb{G}$ as well as $\phi_1, \phi_2 \xleftarrow{R} \mathbb{Z}_p$ and define vectors

$$\boldsymbol{f}_1 = (f_1, 1, g), \qquad \boldsymbol{f}_2 = (1, f_2, g) \qquad \boldsymbol{f}_3 = \boldsymbol{f}_1^{\phi_1} \cdot \boldsymbol{f}_2^{\phi_2} \cdot (1, 1, g).$$

4. Choose random polynomials $P_1[Z], P_2[Z], P_0[Z] \in \mathbb{Z}_p[Z]$ of degree $t-1$ such that $P_1(0) = x_1$, $P_2(0) = x_2$ and $P_0(0) = x_0$. Set $X_{i,1} = f^{P_1(i)} g^{P_0(i)}$ and $X_{i,2} = h^{P_2(i)} g^{P_0(i)}$ for $i = 1$ to N.
5. Choose a collision-resistant hash function $H : \{0,1\}^* \to \mathbb{Z}_p$.
6. Generate a key pair for the one-time linearly homomorphic signature of Section 2.5 with $n = 7$. Let $\big(g_z, g_r, h_z, h_u, \{(g_i, h_i)\}_{i=1}^7\big)$ be the public key and let $\{(\chi_i, \gamma_i, \delta_i)\}_{i=1}^7$ be the corresponding private key.
7. Generate one-time homomorphic signatures $\{(Z_j, R_j, U_j)\}_{j=1}^4$ on the independent vectors

$$\boldsymbol{h}_1 = (f, 1, g, Y_1, 1, 1, 1) \in \mathbb{G}^7, \qquad \boldsymbol{h}_2 = (1, h, g, Y_2, 1, 1, 1) \in \mathbb{G}^7$$
$$\boldsymbol{h}_3 = (1, 1, 1, W_1, f, 1, g) \in \mathbb{G}^7, \qquad \boldsymbol{h}_4 = (1, 1, 1, W_2, 1, h, g) \in \mathbb{G}^7.$$

8. Define decryption key shares $\mathbf{SK} = (SK_1, \ldots, SK_N)$ as $SK_i = (P_1(i), P_2(i), P_0(i)) \in \mathbb{Z}_p^3$ for each $i \in \{1, \ldots, N\}$. The vector $\mathbf{VK} = (VK_1, \ldots, VK_N)$ of verification keys is defined as $VK_i = (X_{i,1}, X_{i,2}) \in \mathbb{G}^2$ for each $i \in \{1, \ldots, N\}$. The public key is defined to be

$$PK = \Big(g,\ \boldsymbol{f}_1,\ \boldsymbol{f}_2,\ \boldsymbol{f}_3,\ X_1,\ X_2,\ Y_1,\ Y_2,\ W_1,\ W_2,\ g_z,\ g_r,\ h_z,\ h_u, \{g_i, h_i\}_{i=1}^7,\ \{(Z_j, R_j, U_j)\}_{j=1}^4,\ H \Big).$$

Encrypt(M, PK)**:** to encrypt a message $M \in \mathbb{G}$, conduct the following steps.

1. Choose $\theta_1, \theta_2 \xleftarrow{R} \mathbb{Z}_p$ and compute

$$C_0 = M \cdot X_1^{\theta_1} \cdot X_2^{\theta_2}, \quad C_1 = f^{\theta_1}, \quad C_2 = h^{\theta_2}, \quad C_3 = g^{\theta_1 + \theta_2} \quad C_4 = (W_1^\alpha Y_1)^{\theta_1} \cdot (W_2^\alpha Y_2)^{\theta_2},$$

   where $\alpha = H(C_0, C_1, C_2, C_3) \in \mathbb{Z}_p$.
2. Construct a linearly homomorphic signature (Z, R, U) on $(C_1, C_2, C_3, C_4, C_1^\alpha, C_2^\alpha, C_3^\alpha) \in \mathbb{G}^7$. Namely, compute

$$Z = Z_1^{\theta_1} \cdot Z_2^{\theta_2} \cdot Z_3^{\theta_1 \cdot \alpha} \cdot Z_4^{\theta_2 \cdot \alpha}, \qquad R = R_1^{\theta_1} \cdot R_2^{\theta_2} \cdot R_3^{\theta_1 \cdot \alpha} \cdot R_4^{\theta_2 \cdot \alpha}, \qquad U = U_1^{\theta_1} \cdot U_2^{\theta_2} \cdot U_3^{\theta_1 \cdot \alpha} \cdot U_4^{\theta_2 \cdot \alpha}$$

3. Output the ciphertext

$$C = (C_0, C_1, C_2, C_3, C_4, Z, R, U) \in \mathbb{G}^8 \tag{21}$$

Ciphertext-Verify$\big(PK, C\big)$**:** parse C as per (21). Compute $\alpha = H(C_0, C_1, C_2, C_3) \in \mathbb{Z}_p$ and return 1 if and only if

$$1_{\mathbb{G}_T} = e(g_z, Z) \cdot e(g_r, R) \cdot \prod_{i=1}^{3} e(g_i \cdot g_{i+4}^{\alpha}, C_i) \cdot e(g_4, C_4)$$

$$1_{\mathbb{G}_T} = e(h_z, Z) \cdot e(h_u, U) \cdot \prod_{i=1}^{3} e(h_i \cdot h_{i+4}^{\alpha}, C_i) \cdot \cdot e(h_4, C_4),$$

Share-Decrypt(PK, i, SK_i, C)**:** on inputs $SK_i = (P_1(i), P_2(i), P_0(i)) \in \mathbb{Z}_p^3$ and C, return $(i, \perp)$ in the event that **Ciphertext-Verify**$\big(PK, C\big) = 0$. Otherwise, compute $\hat{\mu}_i = \big(\nu_i, \boldsymbol{C}_{P_1}, \boldsymbol{C}_{P_2}, \boldsymbol{C}_P, \pi_{\mu_i}\big)$ which consists of a partial decryption $\nu_i = C_1^{P_1(i)} \cdot C_2^{P_2(i)} \cdot C_3^{P_0(i)}$, commitments $\boldsymbol{C}_{P_1}, \boldsymbol{C}_{P_2}, \boldsymbol{C}_{P_0}$ to exponents $P_1(i), P_2(i), P_0(i) \in \mathbb{Z}_p$ and a proof π_{ν_i} that these satisfy the equations

$$\nu_i = C_1^{P_1(i)} \cdot C_2^{P_2(i)} \cdot C_3^{P_0(i)}, \qquad X_{i,1} = f^{P_1(i)} g^{P_0(i)}, \qquad X_{i,2} = h^{P_2(i)} g^{P_0(i)}. \tag{22}$$

The commitments $\boldsymbol{C}_{P_1}, \boldsymbol{C}_{P_2}, \boldsymbol{C}_{P_0}$ and the proof π_{ν_i} are generated using the CRS $\mathbf{f} = (\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{f}_3)$.

Share-Verify$\big(PK, VK_i, C, (i, \hat{\mu}_i)\big)$**:** parse the ciphertext C as $(C_0, C_1, C_2, C_3, C_4, Z, R, U)$ and VK_i as $(X_{i,1}, X_{i,2}) \in \mathbb{G}^2$. If $\hat{\mu}_i = \perp$ or $\hat{\mu}_i$ cannot be properly parsed as $\big(\nu_i, \boldsymbol{C}_{P_1}, \boldsymbol{C}_{P_2}, \boldsymbol{C}_{P_0}, \pi_{\mu_i}\big)$, return 0. Otherwise, return 1 if π_{μ_i} is a valid proof. In any other situation, return 0.

Combine$(PK, \mathbf{VK}, C, \{(i, \hat{\mu}_i)\}_{i \in S})$**:** for each $i \in S$, parse the share $\hat{\mu}_i$ as $\big(\nu_i, \boldsymbol{C}_{P_1}, \boldsymbol{C}_{P_2}, \boldsymbol{C}_P, \pi_{\mu_i}\big)$ and return $\perp$ if **Share-Verify**$\big(PK, C, (i, \hat{\mu}_i)\big) = 0$. Otherwise, compute $\nu = \prod_{i \in S} \nu_i^{\Delta_{i,S}(0)}$, which equals $\nu = C_1^{x_1} \cdot C_2^{x_2} \cdot C_3^{x_0} = X_1^{\theta_1} \cdot X_2^{\theta_2}$ and in turn reveals $M = C_0/\nu$.

If each element has a 256-bit representation on BN curves [4] at the 128-bit security level, the ciphertext overhead amounts to 1792 bits. The DLIN-based scheme of [42] has a ciphertext overhead comprised of 14 group elements and a one-time signature with its verification key (or 4864 bits using Groth's one-time signature [32]). The results of Escala *et al.* [25] reduce this overhead to 3328 bits. The recent techniques of Jutla and Roy [37, 38] – which also work in the threshold setting although it was not explicitly stated in [37] – lead to ciphertexts comprised of 9 group elements under the DLIN assumption and $3k + 3$ under the k-linear assumption. Under DLIN, we thus further compress ciphertexts by 11% while relying on the same assumption and retaining tight security[7].

Under the k-linear assumption, our improvement becomes more important as the ciphertext reduces to $2k + 4$ group elements. Specifically, we need $k + 1$ elements for the homomorphic signature of Appendix D, another $k + 1$ elements to contain the k-linear instance, one element for the Cramer-Shoup-like proof π_0 and one element to carry the plaintext. This allows saving $k - 1$ group elements with respect to the techniques of [37, 38].

We believe this result to be of importance as these schemes can potentially serve as building blocks for protocols in the multi-linear setting [29, 19]. Indeed, the $(k - 1)$-linear problem is easy in groups equipped with a k-linear map (as shown in, *e.g.*, [25]) but we can hope for instantiations where the k-linear assumption holds, as seems to be the case in [19].

From a computational standpoint, the validity of a ciphertext only requires to compute a product of 7 pairings. Under the the DLIN assumption, the framework of [42] requires a product of 12 pairings in the ciphertext verification algorithm.

[7] Note that the techniques of Lewko [40] can be applied to the scheme of [41] to get a DLIN-based system where ciphertexts contain 7 group elements and a one-time key pair (SVK, σ). However, the reduction involves a degradation factor proportional to the number of decryption queries.

I.2 Security

We prove security in the sense of a definition which is identical to Definition 1 with the difference that there is no evaluation key SK_h, no evaluation oracle and no RevHK oracle.

As in the scheme of [37], the security proof appeals to the private verification algorithm while the scheme itself only uses the public verification algorithm.

While it would be possible to rely on the relative zero-knowledge and relative soundness properties of the proof system in a modular way, we obtain a better exact security via a direct analysis.

Theorem 4. *The above threshold cryptosystem is IND-CCA secure against adaptive corruptions assuming that: (i) H is collision-resistant; (ii) The DLIN assumption holds in $\mathbb{G}$. More precisely, the advantage of any PPT adversary $\mathcal{A}$ is at most*

$$\mathbf{Adv}(\mathcal{A}) \leq \mathbf{Adv}^{\text{CR-hash}}(\lambda) + 3 \cdot \mathbf{Adv}^{\text{DLIN}}(\lambda) + \frac{2q+1}{2^{\lambda}-q}, \tag{23}$$

where q is the number of decryption queries and the first term of the right-hand-side member accounts for the maximal advantage of any PPT collision-finding algorithm for H.

Proof. The proof uses of a sequence of games starting with the real attack game and ending with a game where the adversary $\mathcal{A}$ has no advantage. For each i, S_i stands for the event that the challenger $\mathcal{B}$ outputs 1 at the end of Game_i.

Game_1**:** is the real attack game. In details, the adversary is given the public key PK and the set of verification keys $\mathbf{VK} = (VK_1, \ldots, VK_N)$. At each corruption query $i \in \{1, \ldots, n\}$, the challenger $\mathcal{B}$ reveals the queried private key share $SK_i = (P_1(i), P_2(i), P_0(i))$. At each decryption query, $\mathcal{B}$ faithfully runs the real shared decryption algorithm. In the challenge phase, the adversary $\mathcal{A}$ chooses messages $M_0, M_1 \in \mathbb{G}$ and obtains $C^\star = (C_0^\star, C_1^\star, C_2^\star, C_3^\star, C_4^\star, Z^\star, R^\star, U^\star)$ which is an encryption of M_β, for some random coin $\beta \xleftarrow{R} \{0,1\}$ flipped by $\mathcal{B}$. We can assume w.l.o.g. that $(C_1^\star, C_2^\star, C_3^\star)$ are computed at the beginning of the game as they do not depend on M_β.

In the second phase, $\mathcal{A}$ makes more adaptive queries under the restriction of not asking for a partial decryption of $C^\star$ or for more than $t-1$ private key shares throughout the entire game. Eventually, $\mathcal{A}$ halts and outputs β'. At this point, $\mathcal{B}$ outputs 1 if $\beta = \beta'$ and 0 otherwise.

Game_2**:** This game is like Game_1 except that the challenger $\mathcal{B}$ halts and outputs a random bit in the event that, before the challenger phase, $\mathcal{A}$ queries the partial decryption oracle on a ciphertext $C = (C_0, C_1, C_2, C_3, C_4, Z, R, U)$ such that $(C_1, C_2, C_3) = (C_1^\star, C_2^\star, C_3^\star)$. Since $(C_1^\star, C_2^\star, C_3^\star)$ are invisible to $\mathcal{A}$ until the challenge phase, this event can only occur with probability q/p, so that $|\Pr[S_2] - \Pr[S_1]| < q/p$.

Game_3**:** We introduce another failure event F_3 and let $\mathcal{B}$ halt and output a random bit if this event occurs. We define F_3 as the event that $\mathcal{A}$ makes a decryption query involving a valid ciphertext $C = (C_0, C_1, C_2, C_3, C_4, Z, R, U)$ such that $H(C_0, C_1, C_2, C_3) = H(C_0^\star, C_1^\star, C_2^\star, C_3^\star)$ but $(C_0^\star, C_1^\star, C_2^\star, C_3^\star) \neq (C_0, C_1, C_2, C_3)$.

We see that Game_3 and Game_2 are identical until event F_3 occurs, which would contradict the collision-resistance of H. We thus have $|\Pr[S_3] - \Pr[S_2]| \leq \Pr[F_3] \leq \mathbf{Adv}^{\text{CR-hash}}(\lambda)$. In subsequent games, if we define the values $\alpha = H(C_0, C_1, C_2, C_3)$ and $\alpha^\star = H(C_0^\star, C_1^\star, C_2^\star, C_3^\star)$, we will henceforth assume that $\alpha \neq \alpha^\star$ for each decryption query $C = (C_0, C_1, C_2, C_3, C_4, Z, R, U)$.

Game_4**:** In this game, we modify the decryption oracle and reject all post-challenge decryption queries $(C_0, C_1, C_2, C_3, C_4, Z, R, U)$ such that $(C_0, C_1, C_2, C_3, C_4) = (C_0^\star, C_1^\star, C_2^\star, C_3^\star, C_4^\star)$. Clearly Game_4 is identical to Game_3 until $\mathcal{B}$ rejects a ciphertext that would not have been rejected in Game_3.

If we call the latter event F_4, we find that $|\Pr[S_4] - \Pr[S_3]| \leq \Pr[F_4]$. Since F_4 necessarily implies $(Z, R, U) \neq (Z^\star, R^\star, U^\star)$, any occurrence of F_4 necessarily provides $\mathcal{A}$ with two distinct

signatures on the same vector $(C_1^\star, C_2^\star, C_3^\star, C_4^\star, C_1^{\star\alpha^\star}, C_2^{\star\alpha^\star}, C_3^{\star\alpha^\star})$, which in turn breaks the SDP assumption by the specific property of the linearly homomorphic signature (see Section 2.5). It comes that $\Pr[F_4] \leq \mathbf{Adv}^{\mathrm{SDP}}(\mathcal{B})$.

Game_5**:** We modify the generation of $C^\star = (C_0^\star, C_1^\star, C_2^\star, C_3^\star, C_4^\star, Z^\star, R^\star, U^\star)$ in the challenge phase. Specifically, instead of computing $C_0^\star = M_\beta \cdot X_1^{\theta_1} \cdot X_2^{\theta_2}$ and $C_4^\star = (W_1^{\alpha^\star} Y_1)^{\theta_1} (W_2^{\alpha^\star} Y_2)^{\theta_2}$, where $\theta_1 = \log_f(C_1^\star)$ and $\theta_2 = \log_h(C_2^\star)$, the challenger $\mathcal{B}$ now computes $C_0^\star = M_\beta \cdot C_1^{\star x_1} \cdot C_2^{\star x_2} \cdot C_3^{\star x_0}$ and $C_4^\star = C_1^{\star y_1 + \alpha^\star w_1} \cdot C_2^{\star y_2 + \alpha^\star w_2} \cdot C_3^{\star y_0 + \alpha^\star w_0}$, with $\alpha^\star = H(C_0^\star, C_1^\star, C_2^\star, C_3^\star)$. Likewise, instead of using the encryption exponents (θ_1, θ_2) to derive a one-time linearly homomorphic signature $(Z^\star, R^\star, U^\star)$ from the public key, the challenger $\mathcal{B}$ uses $\{(\chi_i, \gamma_i, \delta_i)\}_{i=1}^7$ and computes

$$Z^\star = \prod_{i=1}^{7} C_i^{\star -\chi_i} \qquad R^\star = \prod_{i=1}^{7} C_i^{\star -\gamma_i} \qquad U^\star = \prod_{i=1}^{7} C_i^{\star -\delta_i}, \tag{24}$$

where $(C_5^\star, C_6^\star, C_7^\star) = (C_1^{\star\alpha}, C_2^{\star\alpha}, C_3^{\star\alpha})$.

This change is only conceptual since $C_0^\star$ still equals $C_0^\star = M_\beta \cdot X_1^{\theta_1} \cdot X_2^{\theta_2}$ and the distribution of $(C_4^\star, Z^\star, R^\star, U^\star)$ has not changed either. We thus have $\Pr[S_5] = \Pr[S_4]$.

Game_6**:** Here, we modify the decryption oracle and make use of the exponents $(y_0, y_1, y_2, w_0, w_1, w_2)$ that were chosen by $\mathcal{B}$ during the key generation phase. Namely, the challenger $\mathcal{B}$ does not only reject all ciphertexts $(C_0, C_1, C_2, C_3, C_4, Z, R, U)$ such that (Z, R, U) does not form a valid signature on $(C_1, C_2, C_3, C_4, C_1^\alpha, C_2^\alpha, C_3^\alpha)$ but also rejects those for which

$$C_4 \neq C_1^{y_1 + \alpha \cdot w_1} \cdot C_2^{y_2 + \alpha \cdot w_2} \cdot C_3^{y_0 + \alpha \cdot w_0},$$

where $\alpha = H(C_0, C_1, C_2, C_3)$. We raise a failure event F_6, which causes $\mathcal{B}$ to halt and output a random bit if it occurs. This event F_6 is defined to be the event that the adversary $\mathcal{A}$ queries the decryption oracle on a ciphertext that gets rejected in Game_6 and would not have been rejected in Game_5. Since Game_6 is identical to Game_5 until F_6 occurs, we have

$$\Pr[S_6] = \Pr[S_6 \wedge \neg F_6] + \frac{1}{2} \cdot \Pr[F_6] = \Pr[S_5] + \frac{1}{2} \cdot \Pr[F_6].$$

At the same time, Lemma 3 shows that $\Pr[F_6] \leq \mathbf{Adv}^{\mathrm{SDP}}(\lambda) + \frac{1}{p}$. We remark that a side-effect of this modified decryption oracle is that it now rejects all post-challenge decryption queries $(C_0, C_1, C_2, C_3, C_4, Z, R, U)$ such that $(C_0, C_1, C_2, C_3) = (C_0^\star, C_1^\star, C_2^\star, C_3^\star)$ but $C_4 \neq C_4^\star$.

Since F_6 is always efficiently detectable by the challenger $\mathcal{B}$, we can build an efficient DLIN distinguisher if the probability of event F_6 increases when $C_3^\star$ is tampered with in the challenge ciphertext as in the next game.

Game_7**:** This game is identical to Game_6 with one modification in the challenge ciphertext. Instead of setting $C_3^\star = g^{\theta_1 + \theta_2}$, where $\theta_1 = \log_f(C_1^\star)$ and $\theta_2 = \log_h(C_2^\star)$, we choose it as $C_3^\star \xleftarrow{R} \mathbb{G}$. The linearly homomorphic signature $(Z^\star, R^\star, U^\star)$ is computed according to (24), as previously. Under the DLIN assumption in $\mathbb{G}$, this modification should have no noticeable impact on $\mathcal{A}$'s behavior. In particular, we have $|\Pr[S_7] - \Pr[S_6]| \leq \mathbf{Adv}^{\mathrm{DLIN}}(\lambda)$.

Game_8**:** We modify the partial decryption oracle and replace the non-interactive proofs contained in decryption shares $\hat{\mu}_i$ by simulated NIZK proofs. This entails to turn $(\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{f}_3)$ into a perfectly hiding Groth-Sahai CRS (where $\boldsymbol{f_3}$ is in $\mathrm{span}(\boldsymbol{f_1}, \boldsymbol{f_2})$) and non-interactive proofs for multi-exponentiation equations are simulated using the trapdoor of the simulated CRS. Under the DLIN assumption, this change is not noticeable by $\mathcal{A}$ and we have $|\Pr[S_8] - \Pr[S_7]| \leq \mathbf{Adv}^{\mathrm{DLIN}}(\lambda)$.

Game_9**:** In this game, we modify again the decryption oracle and make use of the discrete logarithms $\alpha_f = \log_g(f)$ and $\alpha_h = \log_g(h)$. Since we are done with the transition consisting in replacing $C_3^\star$ by a random element, we are free to use (α_f, α_h) from this point forward. We thus introduce

a modification in the treatment of decryption queries $C = (C_0, C_1, C_2, C_3, C_4, Z, R, U)$. This, $\mathcal{B}$ rejects *all* ciphertexts C such that $C_3 \neq C_1^{1/\alpha_f} \cdot C_2^{1/\alpha_h}$. Otherwise, it answers as in Game_8.

If we define F_9 to be the event that $\mathcal{B}$ rejects a ciphertext which would not have been rejected in Game_8, we see that Game_9 and Game_8 are identical from $\mathcal{A}$'s view until F_9 occurs. Therefore it comes that

$$\Pr[S_9] \leq \Pr[S_9 \wedge \neg F_9] + \Pr[F_9] = \Pr[S_8] + \Pr[F_9].$$

The same arguments as in the proof of Cramer and Shoup show that $\Pr[F_9] \leq q/(p-q)$. More precisely, after i decryption queries, the adversary is left with $p - i$ equally likely candidates for the value of C_4 that would have been accepted by the private ciphertext validation algorithm. The probability that the i-th decryption query satisfies the test given that the first $i-1$ queries have failed it is thus at most $i/(p-i)$.

In Game_9, it is easy to see that $\mathcal{A}$ has no advantage whatsoever and we have $\Pr[S_9] = 1/2$. Indeed, in the challenge phase, we have $(C_1^\star, C_2^\star, C_3^\star) = (f^{\theta_1}, h^{\theta_2}, g^{\theta_1+\theta_2+\theta_3})$, with $\theta_1, \theta_2, \theta_3 \in_R \mathbb{Z}_p$, so that $C_0^\star$ can be written as $C_0^\star = M_\beta \cdot X_1^{\theta_1} \cdot X_2^{\theta_2} \cdot g^{\theta_3 \cdot x_0}$. The latter equality implies that, as long as $x_0 \in \mathbb{Z}_p$ is independent of $\mathcal{A}$'s view, so is the bit $\beta \in \{0,1\}$.

We also note that, in Game_9, decryption shares $\hat{\mu}_i$ contain NIZK proofs that are simulated without using private key shares and thus leak no information about these. It comes that, as long as $\mathcal{A}$ does not corrupt more than $t-1$ servers, the only possible way to infer information about $x_0 = P(0)$ is to make decryption queries on invalid ciphertexts (*i.e.*, for which (C_1, C_2, C_3) lies outside the span of $\boldsymbol{f}$ and $\boldsymbol{h}$).

We thus find

$$|\Pr[S_1] - \frac{1}{2}| < \mathbf{Adv}^{\text{CR-hash}}(\lambda) + 2 \cdot \mathbf{Adv}^{\text{DLIN}}(\lambda) + 2 \cdot \mathbf{Adv}^{\text{SDP}}(\lambda) + \frac{2q+1}{p-q}.$$

Since any algorithm solving SDP immediately provides a DLIN distinguisher, we also have the inequality $\mathbf{Adv}^{\text{SDP}}(\lambda) \leq \frac{1}{2}\mathbf{Adv}^{\text{DLIN}}(\lambda)$, which yields

$$|\Pr[S_1] - \frac{1}{2}| < \mathbf{Adv}^{\text{CR-hash}}(\lambda) + 3 \cdot \mathbf{Adv}^{\text{DLIN}}(\lambda) + \frac{2q+1}{2^\lambda - q} \tag{25}$$

and the claimed result follows. □

Lemma 3. *In* Game_6*, the probability of event* F_6 *is at most* $\Pr[F_6] \leq \mathbf{Adv}^{\text{SDP}}(\lambda) + \frac{1}{p}$.

Proof. We show that, if event F_6 occurs with non-negligible probability ε in Game_6, there exists an efficient algorithm $\mathcal{B}$ that solves a SDP instance (g_z, g_r, h_z, h_u) with about the same probability. To this end, we first remark that F_6 can only occur for a decryption query $(C_0, C_1, C_2, C_3, C_4, Z, R, U)$ such that $(C_1, C_2, C_3, C_4, C_1^\alpha, C_2^\alpha, C_3^\alpha)$ is outside $\text{span}(\boldsymbol{h_1}, \boldsymbol{h_2}, \boldsymbol{h_3}, \boldsymbol{h_4})$. Indeed, otherwise, there exist integers $\theta_1, \theta_2 \in \mathbb{Z}_p$ such that $(C_1, C_2, C_3, C_4) = \big(f^{\theta_1}, h^{\theta_2}, g^{\theta_1+\theta_2}, (W_1^\alpha Y_1)^{\theta_1}(W_2^\alpha Y_2)^{\theta_2}\big)$, in which case we always have $C_4 = C_1^{y_1+\alpha w_1} \cdot C_2^{y_2+\alpha w_2} \cdot C_3^{y_0+\alpha w_0}$ and the rejection rule of Game_6 does not apply.

Using the technique of [43][Theorem 1], we show that event F_6 implies an algorithm solving the given SDP instance with nearly the same probability. Algorithm $\mathcal{B}$ begins by setting up $g_i = {g_z}^{\chi_i} {g_r}^{\gamma_i}$ and $h_i = {h_z}^{\chi_i} {h_u}^{\delta_i}$, with $\chi_i, \gamma_i, \delta_i \xleftarrow{R} \mathbb{Z}_p$ for $i \in \{1, \ldots, 7\}$. Other public key components are generated as in the real scheme and the public key is given to the adversary.

Throughout the game, the reduction $\mathcal{B}$ answers $\mathcal{A}$'s decryption queries in the same way as in Game_6. By hypothesis, $\mathcal{A}$ must query the decryption of a ciphertext $(C_0, C_1, C_2, C_3, C_4, Z, R, U)$ such that (Z, R, U) is a valid linearly homomorphic signature on the vector $(C_1, C_2, C_3, C_4, C_1^\alpha, C_2^\alpha, C_3^\alpha)$, where $\alpha = H(C_0, C_1, C_2, C_3)$, but $C_4 \neq C_1^{y_1+\alpha w_1} \cdot C_2^{y_2+\alpha w_2} \cdot C_3^{y_0+\alpha w_0}$, which implies that the vector is

not in span($\boldsymbol{h_1}, \boldsymbol{h_2}, \boldsymbol{h_3}, \boldsymbol{h_4}$). When $\mathcal{B}$ detects this event, it defines $(C_5, C_6, C_7) = ({C_1}^{\alpha}, {C_2}^{\alpha}, {C_3}^{\alpha})$ and computes its own signature

$$(Z^{\dagger}, R^{\dagger}, U^{\dagger}) = (\prod_{i=1}^{7} {C_i}^{-\chi_i}, \prod_{i=1}^{7} {C_i}^{-\gamma_i}, \prod_{i=1}^{7} {C_i}^{-\delta_i}) \tag{26}$$

on $(C_1, C_2, C_3, C_4, C_5, C_6, C_7)$. We claim that, with overwhelming probability,

$$(Z^{\ddagger}, R^{\ddagger}, U^{\ddagger}) = \Big(\frac{Z}{Z^{\dagger}}, \frac{R}{R^{\dagger}}, \frac{U}{U^{\dagger}}\Big)$$

is a non-trivial solution to the SDP instance since $Z^{\ddagger} \neq 1_{\mathbb{G}}$ with all but negligible probability.

To see this, we first note that the vector $(\chi_1, \ldots, \chi_7)$ is independent of $\mathcal{A}$'s view before the challenge phase. Hence, since $(C_1, C_2, C_3, C_4, C_5, C_6, C_7)$ is linearly independent of $(\boldsymbol{h}_1, \boldsymbol{h}_2, \boldsymbol{h}_3, \boldsymbol{h}_4)$, the adversary $\mathcal{A}$ can only predict $Z^{\dagger}$ (as it is computed in (26)) with negligible probability $1/p$. The probability $\Pr[F_6]$ can thus be bounded as $\Pr[F_6] \leq \mathbf{Adv}^{\mathrm{SDP}}_{\mathcal{B}}(\lambda) + \frac{1}{p}$. □

In the proof of the above theorem, the relative simulation-soundness property of the proof system is notably used in the transition from Game_8 to Game_9. In order to obtain a tighter reduction, we chose not to rely on this property in a modular way. In the modular approach, we would have to build an algorithm $\mathcal{B}^{rs}$ that contradicts this property using an adversary for which event F_9 occurs with non-negligible probability. This algorithm $\mathcal{B}^{rs}$ would have to interact with the relative soundness challenger for a *given* language $\rho \in \mathbb{G}^{2\times 3}$ for which $\mathcal{B}^{rs}$ does *not* have the underlying matrix $\mathbf{A} \in \mathbb{Z}_p^{2\times 3}$ of discrete logarithms. For this reason, $\mathcal{B}^{rs}$ would not be able to efficiently detect when F_9 occurs. To break the relative soundness property, $\mathcal{B}^{rs}$ would have to guess the decryption query for which this event occurs, which is only possible with probability $1/q$. In the exact security result (23), we would thus lose a multiplicative factor of $O(q)$.